L Number	Hits	Search Text	DB	Time stamp
-	181	380/283.ccls.	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:17
-	149	380/283.ccls. and encrypt$6 with key	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:17
-	19	(380/283.ccls. and encrypt$6 with key) and key with table	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:18
-	23428	encrypt$5 with key	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:18
-	373	(encrypt$5 with key) and "key table"	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:19
-	149	(encrypt$5 with key) and "key table" and (random with number)	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:32
-	3	(encrypt$5 with key) and "key table" and (random with number) and database.ti.	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:28
-	31032	(encrypt$5 with key) and "key table" database.ti.	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:28
-	6	(encrypt$5 with key) and "key table" and database.ti.	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:28
-	292	database with security.ti.	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:34
-	4	database with security.ti. and "encryption key"	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:35
-	38	database with security.ab. and "encryption key"	USPAT; US-PGPUB; EPO; DERWENT	2004/09/22 12:35
-	21	("4531527" | "4588991" | "4712562" | "4838275" | "5012411" | "5313521" | "5418951" | "5553146" | "5606315" | "5706365" | "5772585" | "5832450" | "5845255" | "5862223" | "5864683" | "5868669" | "5903721" | "5903889" | "5940507" | "6092202" | "6230272").PN.	USPAT	2004/09/22 15:06
-	1450	713/200.ccls.	USPAT	2004/09/28 17:00
-	12737	713/$.ccls.	USPAT	2004/09/28 17:01
-	7465	380/$.ccls.	USPAT	2004/09/28 17:01
-	4370	380/$.ccls. and key	USPAT	2004/09/28 17:01

US006785810B1

(12) **United States Patent**
Lirov et al.

(10) **Patent No.: US 6,785,810 B1**
(45) **Date of Patent: Aug. 31, 2004**

(54) **SYSTEM AND METHOD FOR PROVIDING SECURE TRANSMISSION, SEARCH, AND STORAGE OF DATA**

(75) Inventors: **Yuval Lirov**, Morganville, NJ (US); **Erez Lirov**, Morganville, NJ (US)

(73) Assignee: **eSpoc, Inc.**, Morganville, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/388,025**

(22) Filed: **Aug. 31, 1999**

(51) **Int. Cl.**[7] **H04L 9/00**; H04L 9/32; G06F 11/30; G06F 12/14; G06F 17/30; G06F 7/06

(52) **U.S. Cl.** **713/165**; 713/193; 707/9

(58) **Field of Search** 713/165, 193; 707/9

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,531,527 A		7/1985	Reinhold, Jr. et al.	
4,588,991 A	*	5/1986	Atalla	713/165
4,712,562 A		12/1987	Ohayon et al.	
4,838,275 A		6/1989	Lee	
5,012,411 A		4/1991	Policastro et al.	
5,313,521 A	*	5/1994	Torii et al.	380/281
5,418,951 A		5/1995	Damashek	
5,553,146 A	*	9/1996	Flake	713/150
5,606,315 A	*	2/1997	Gaskins	340/5.74
5,706,365 A		1/1998	Rangarajan et al.	
5,772,585 A		6/1998	Lavin et al.	
5,832,450 A		11/1998	Myers et al.	
5,845,255 A		12/1998	Mayaud	
5,862,223 A		1/1999	Walker et al.	
5,864,683 A		1/1999	Boebert et al.	
5,868,669 A		2/1999	Iliff	
5,903,721 A		5/1999	Sixtus	
5,903,889 A		5/1999	de la Huerga et al.	
5,940,507 A	*	8/1999	Cane et al.	713/165
6,092,202 A	*	7/2000	Veil et al.	713/201
6,230,272 B1	*	5/2001	Lockhart et al.	713/202

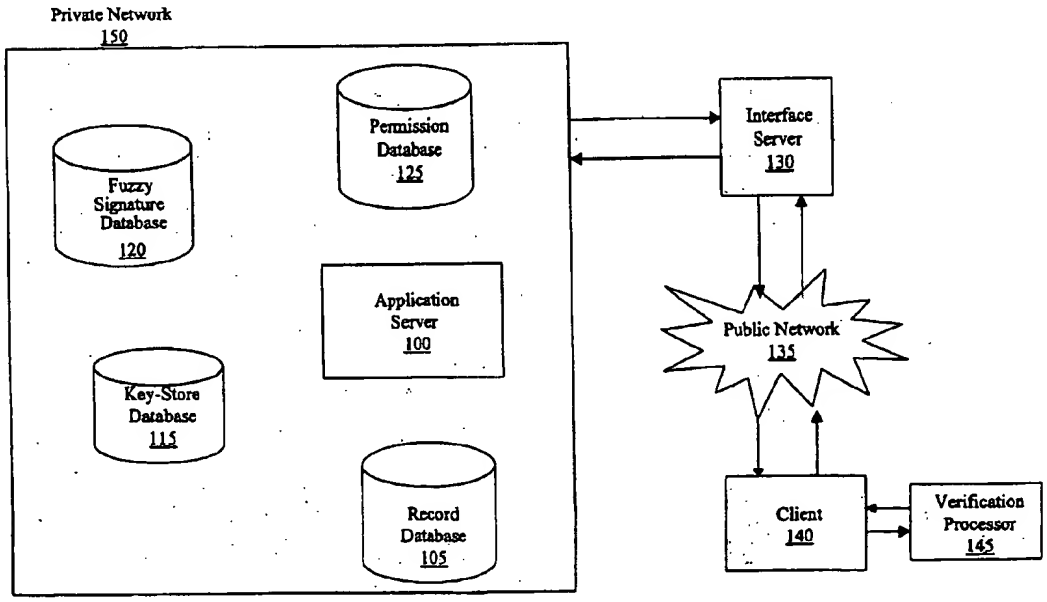
* cited by examiner

Primary Examiner—Gilberto Barron
Assistant Examiner—Benjamin E. Lanier
(74) *Attorney, Agent, or Firm*—Kenyon & Kenyon

(57) **ABSTRACT**

A system and method for securely transmitting, searching, and storing data. To ensure security on the client side of a communication network, the system and method double encrypt sensitive data and single encrypt non-sensitive data. The system and method also fuzzy searches for user information. Thus, it is possible to find the information for the user in a database knowing only a minimal amount of detail about that user. Privacy and security is provided without impeding performance or compromising any of the standard database search functionality. Capitalizing on the difference in privacy requirements between users, the number of keys required to access sensitive data is minimized by using a single key for each user (e.g., a patient) and two keys for other users (e.g., health care providers).

10 Claims, 15 Drawing Sheets



10

Fig. 1a

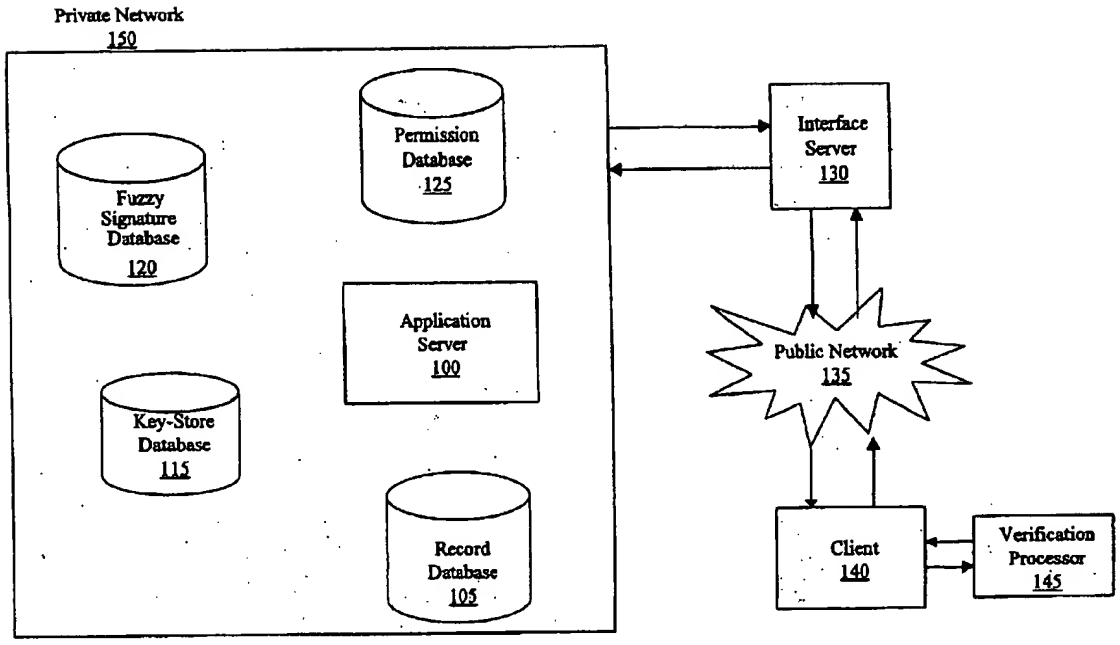

Private Network
150
Permission Database
125
Fuzzy Signature Database
120
Application Server
100
Key-Store Database
115
Record Database
105
Interface Server
130
Public Network
135
Client
140
Verification Processor
145
10

Fig. 1b

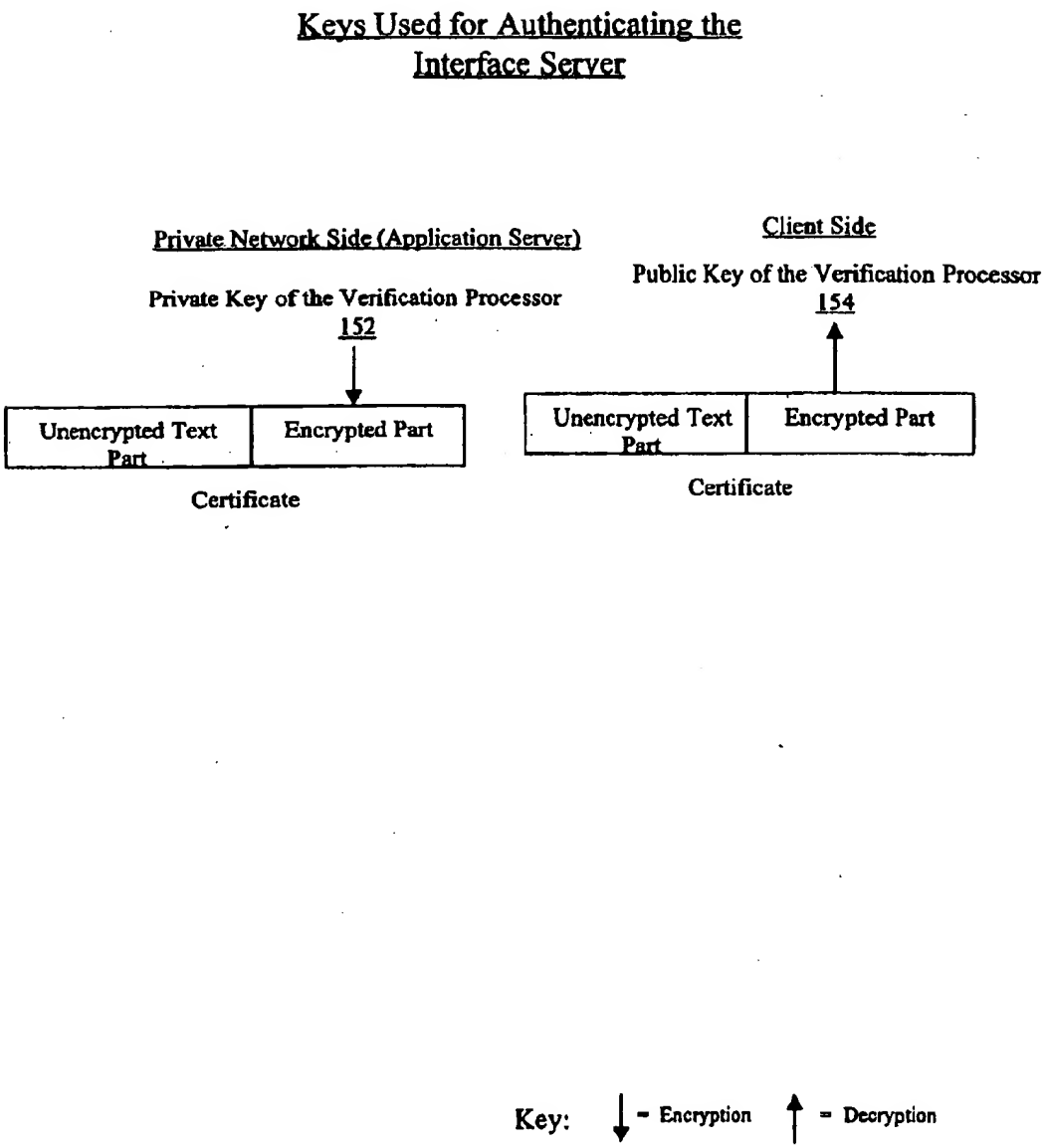

Keys Used for Authenticating the Interface Server
Private Network Side (Application Server)
Private Key of the Verification Processor
152
Unencrypted Text Part
Encrypted Part
Certificate
Client Side
Public Key of the Verification Processor
154
Unencrypted Text Part
Encrypted Part
Certificate
Key: = Encryption = Decryption

Fig. 1c

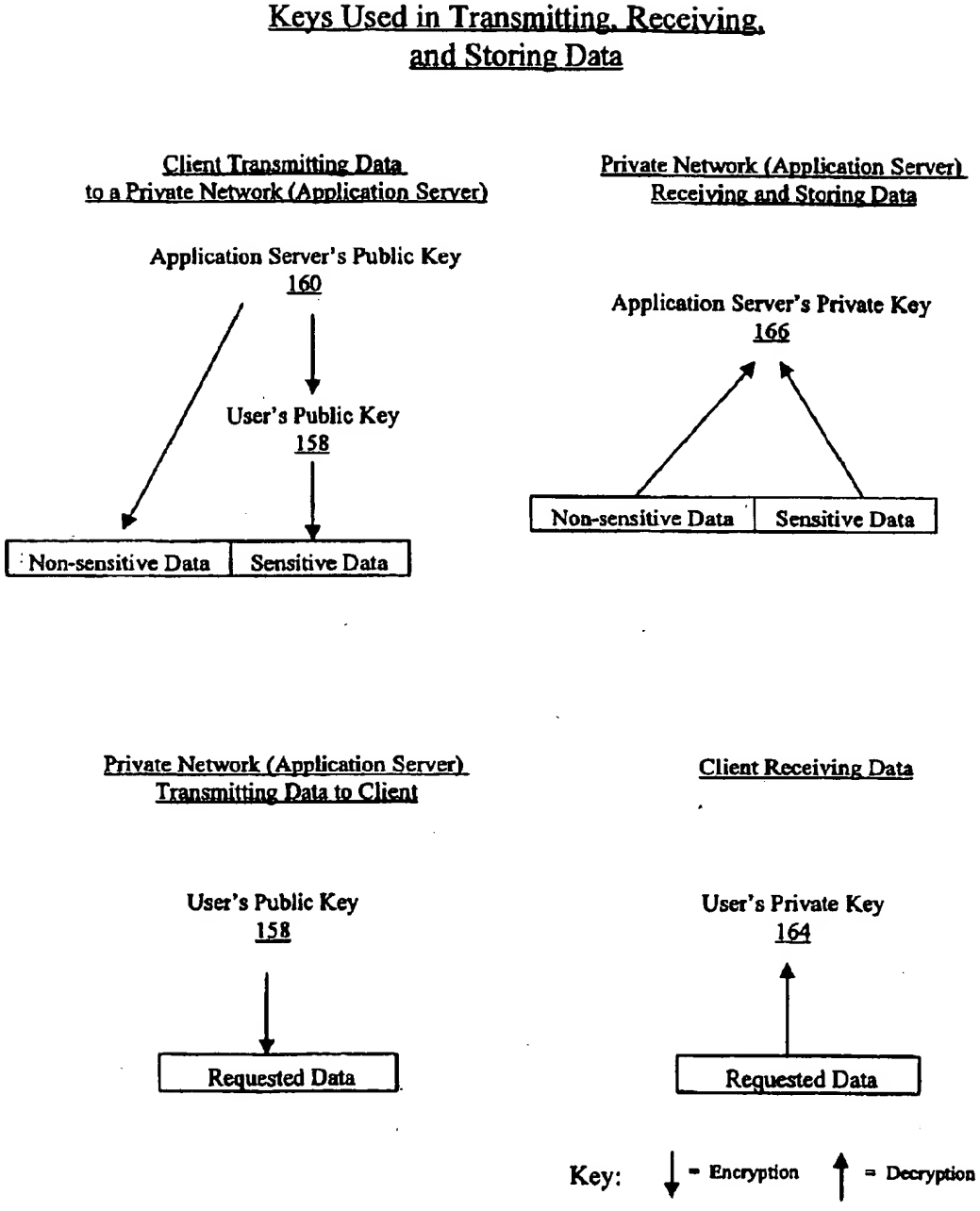
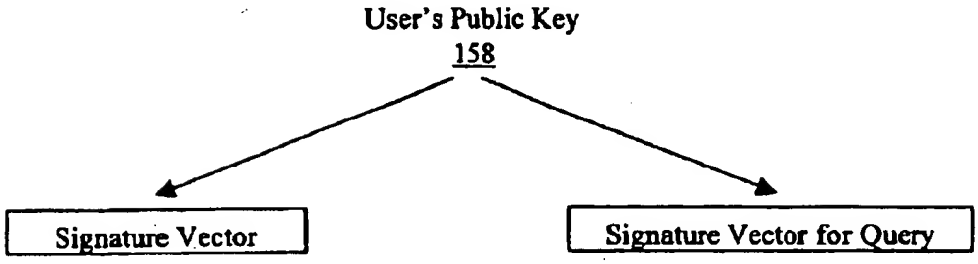

Keys Used in Transmitting, Receiving, and Storing Data
Client Transmitting Data to a Private Network (Application Server)
Application Server's Public Key 160
User's Public Key 158
Non-sensitive Data
Sensitive Data
Private Network (Application Server) Receiving and Storing Data
Application Server's Private Key 166
Non-sensitive Data
Sensitive Data
Private Network (Application Server) Transmitting Data to Client
User's Public Key 158
Requested Data
Client Receiving Data
User's Private Key 164
Requested Data
Key: = Encryption = Decryption

Fig. 1d

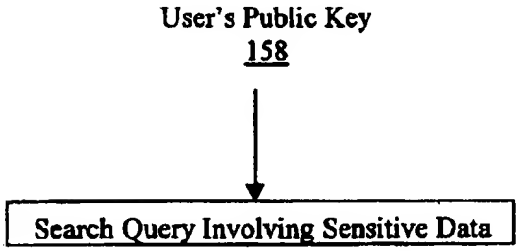
Keys Utilized in Fuzzy Searching



Key: ↓ = Encryption ↑ = Decryption

Fig. 1e

Keys Utilized in Relational Database Searching



Key: ↓ = Encryption ↑ = Decryption

Fig. 1f

Keys Utilized in Accessing Sensitive Data

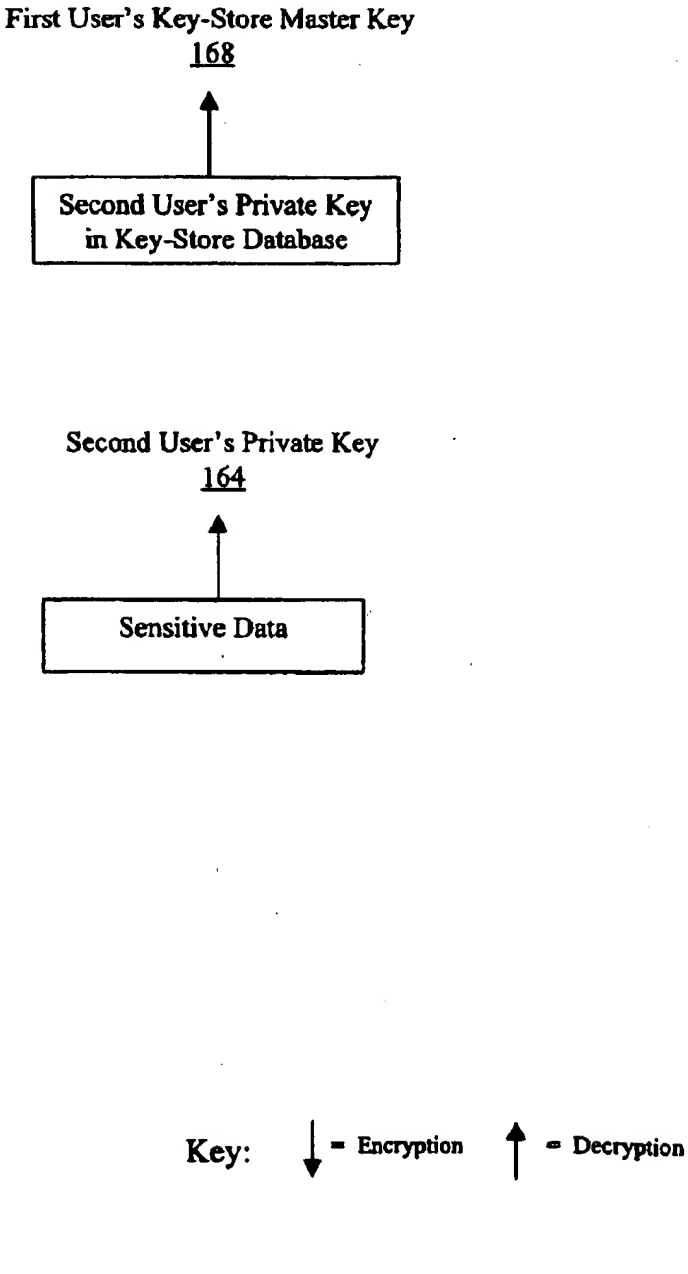


Fig. 1h

Steps for Communications between a Client and a Private Network

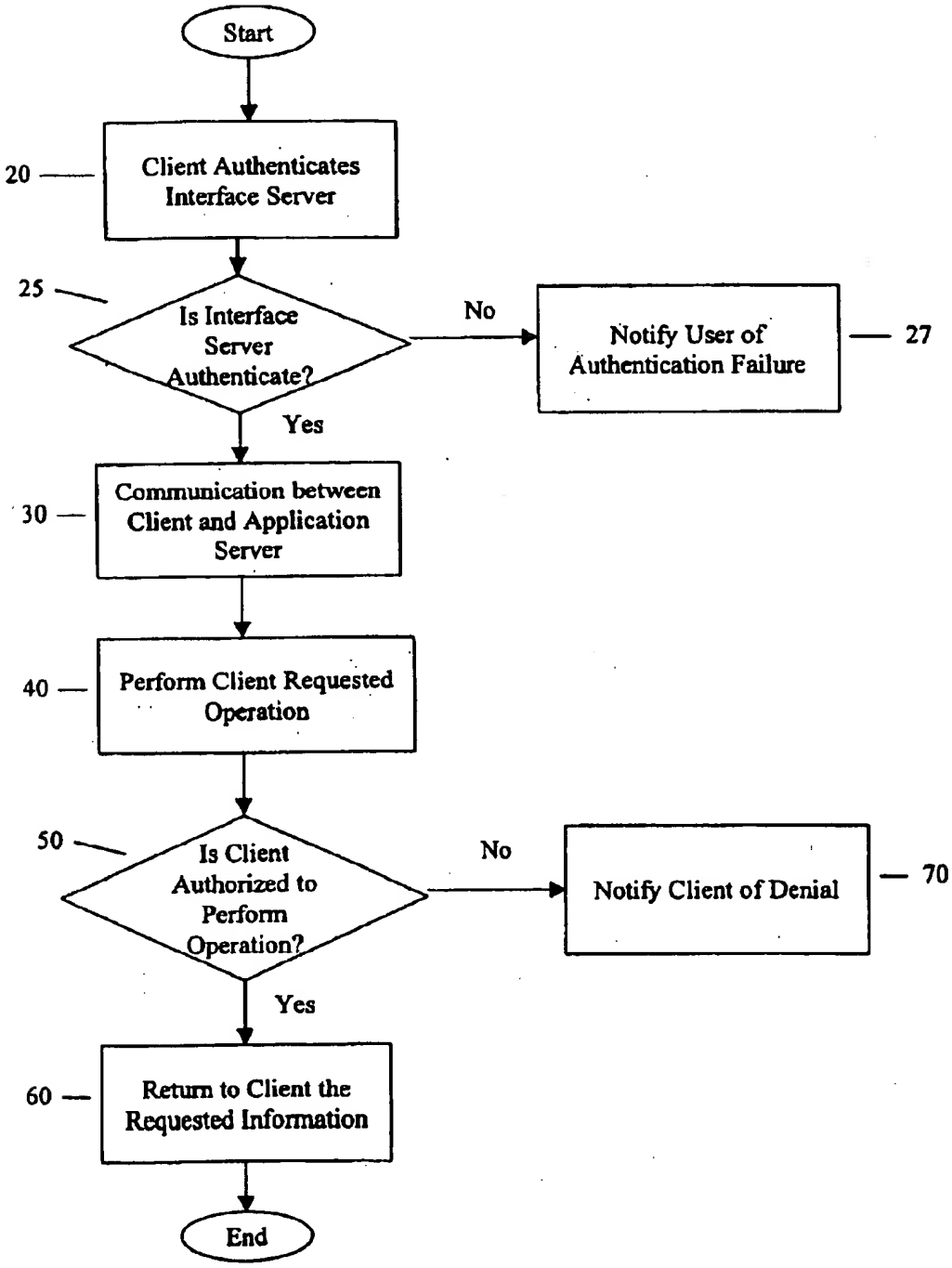


Fig. 2

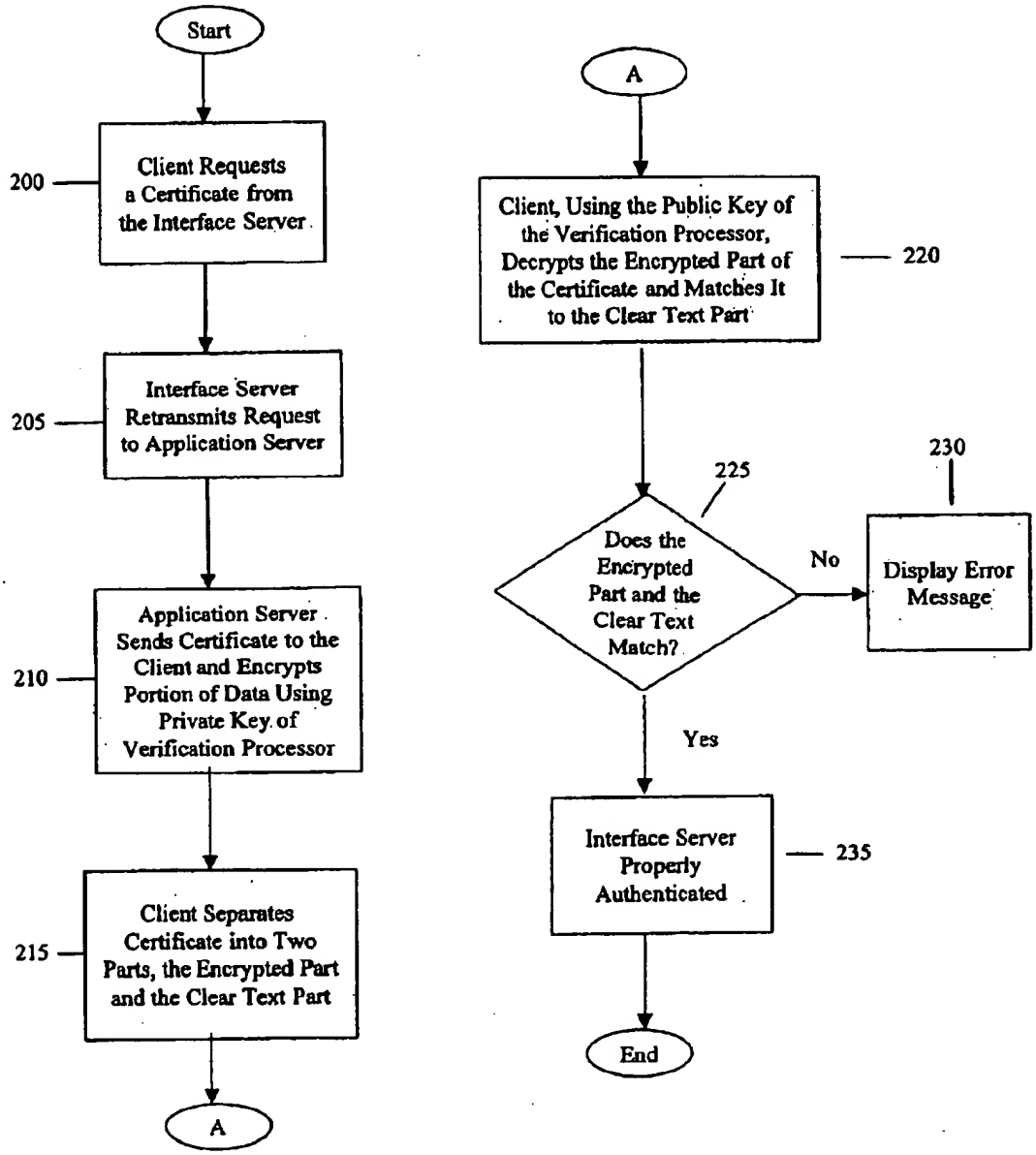

Authentication
Start
200
Client Requests a Certificate from the Interface Server
205
Interface Server Retransmits Request to Application Server
210
Application Server Sends Certificate to the Client and Encrypts Portion of Data Using Private Key of Verification Processor
215
Client Separates Certificate into Two Parts, the Encrypted Part and the Clear Text Part
A
A
Client, Using the Public Key of the Verification Processor, Decrypts the Encrypted Part of the Certificate and Matches It to the Clear Text Part
220
225
Does the Encrypted Part and the Clear Text Match?
No
230
Display Error Message
Yes
Interface Server Properly Authenticated
235
End

Fig. 3

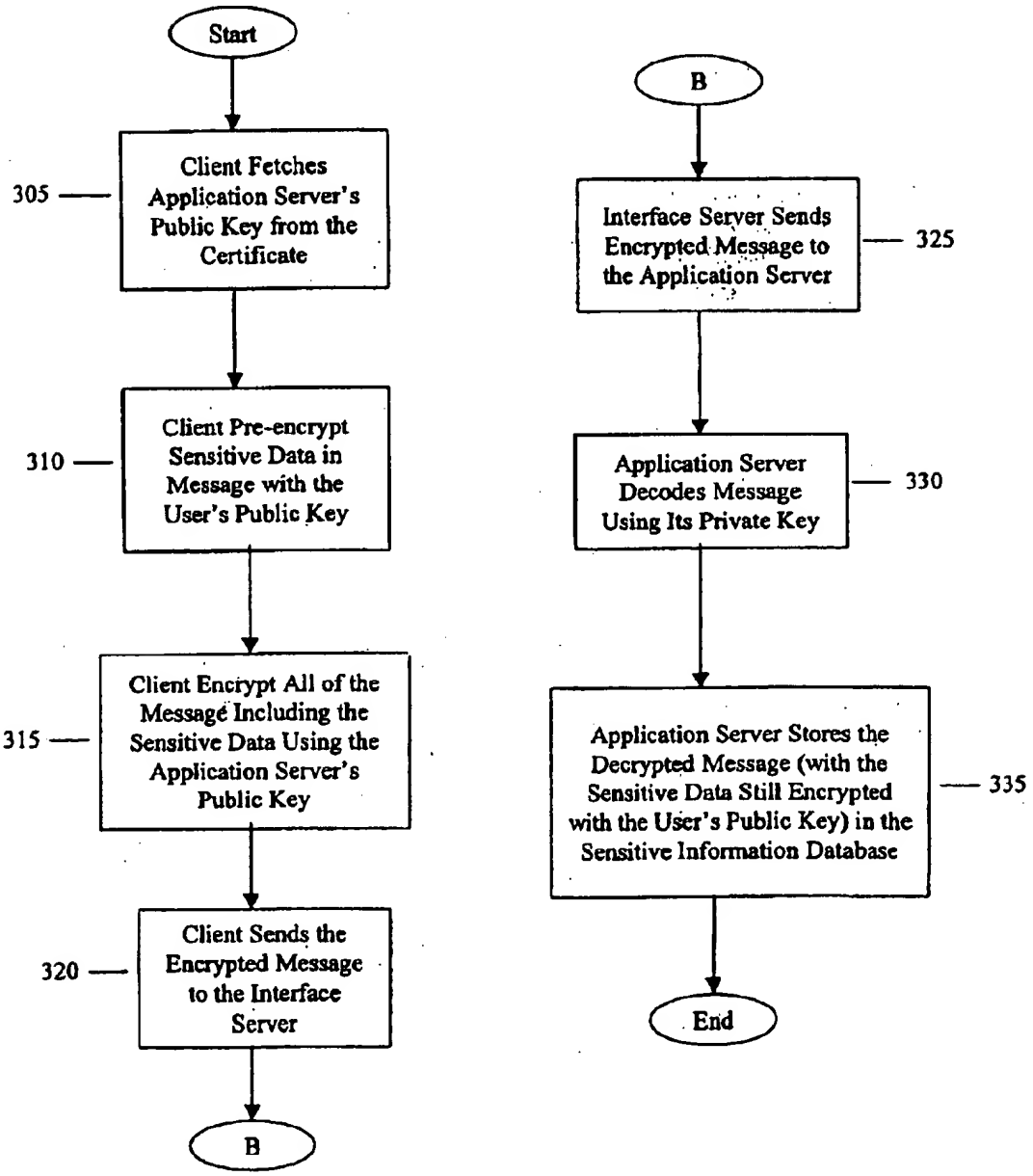

Transmission and Storage of Data
Start
Client Fetches Application Server's Public Key from the Certificate
305
Client Pre-encrypt Sensitive Data in Message with the User's Public Key
310
Client Encrypt All of the Message Including the Sensitive Data Using the Application Server's Public Key
315
Client Sends the Encrypted Message to the Interface Server
320
B
B
Interface Server Sends Encrypted Message to the Application Server
325
Application Server Decodes Message Using Its Private Key
330
Application Server Stores the Decrypted Message (with the Sensitive Data Still Encrypted with the User's Public Key) in the Sensitive Information Database
335
End

FIG. 4

Search Operation

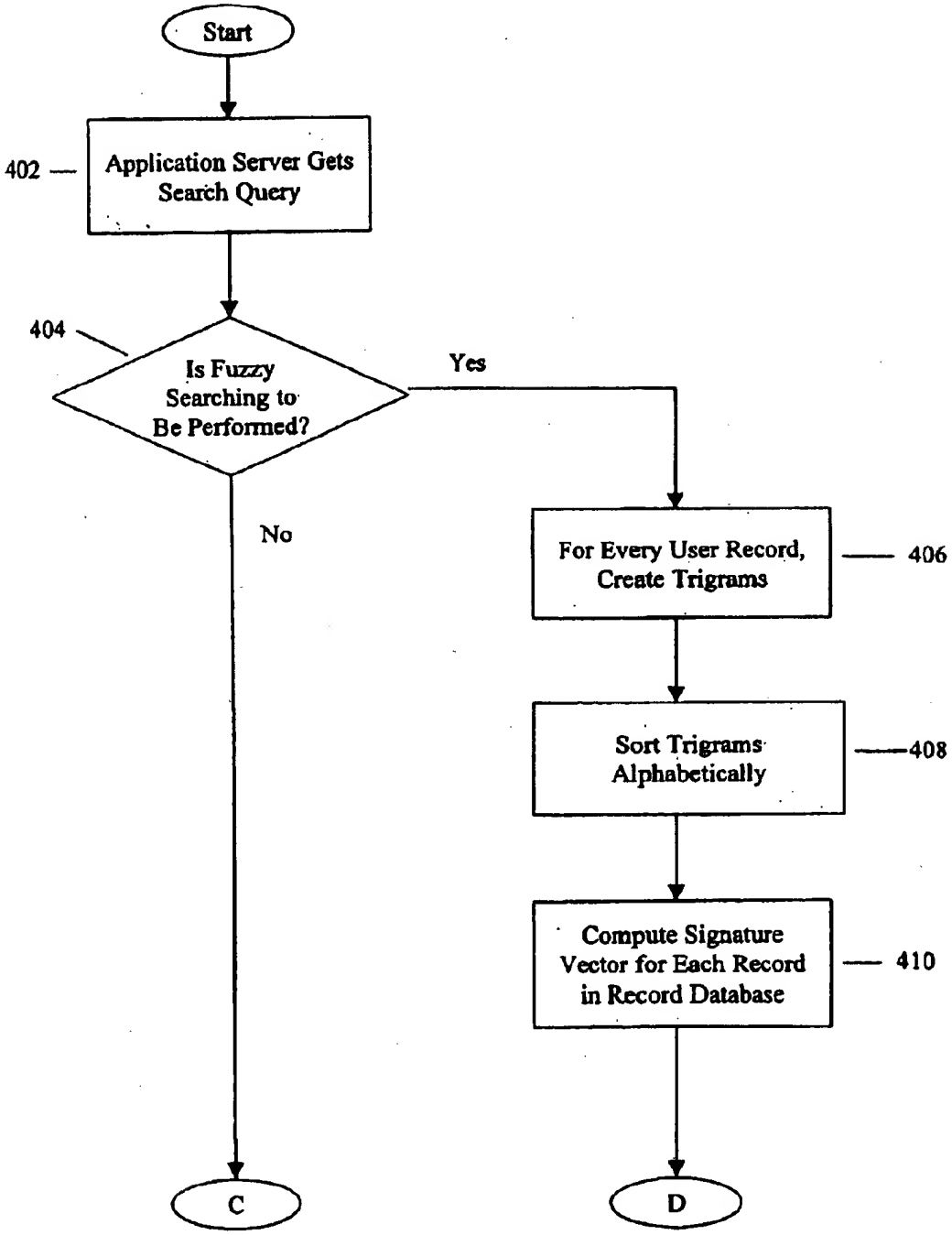


FIG. 4 (continued)

Search Operation

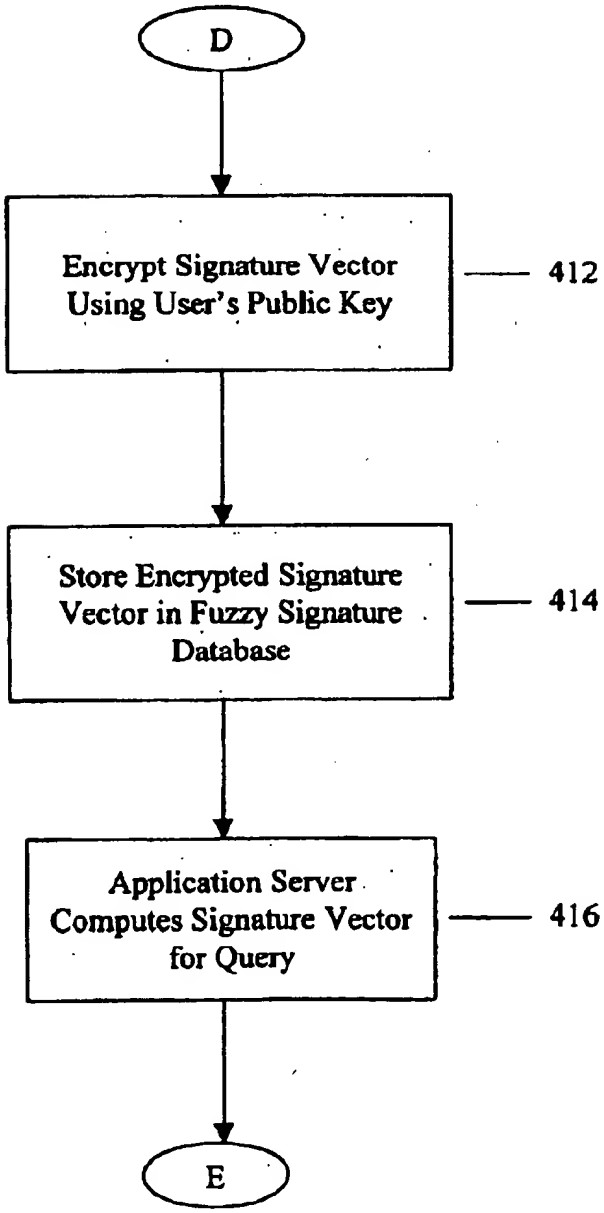


FIG. 4 (continued)

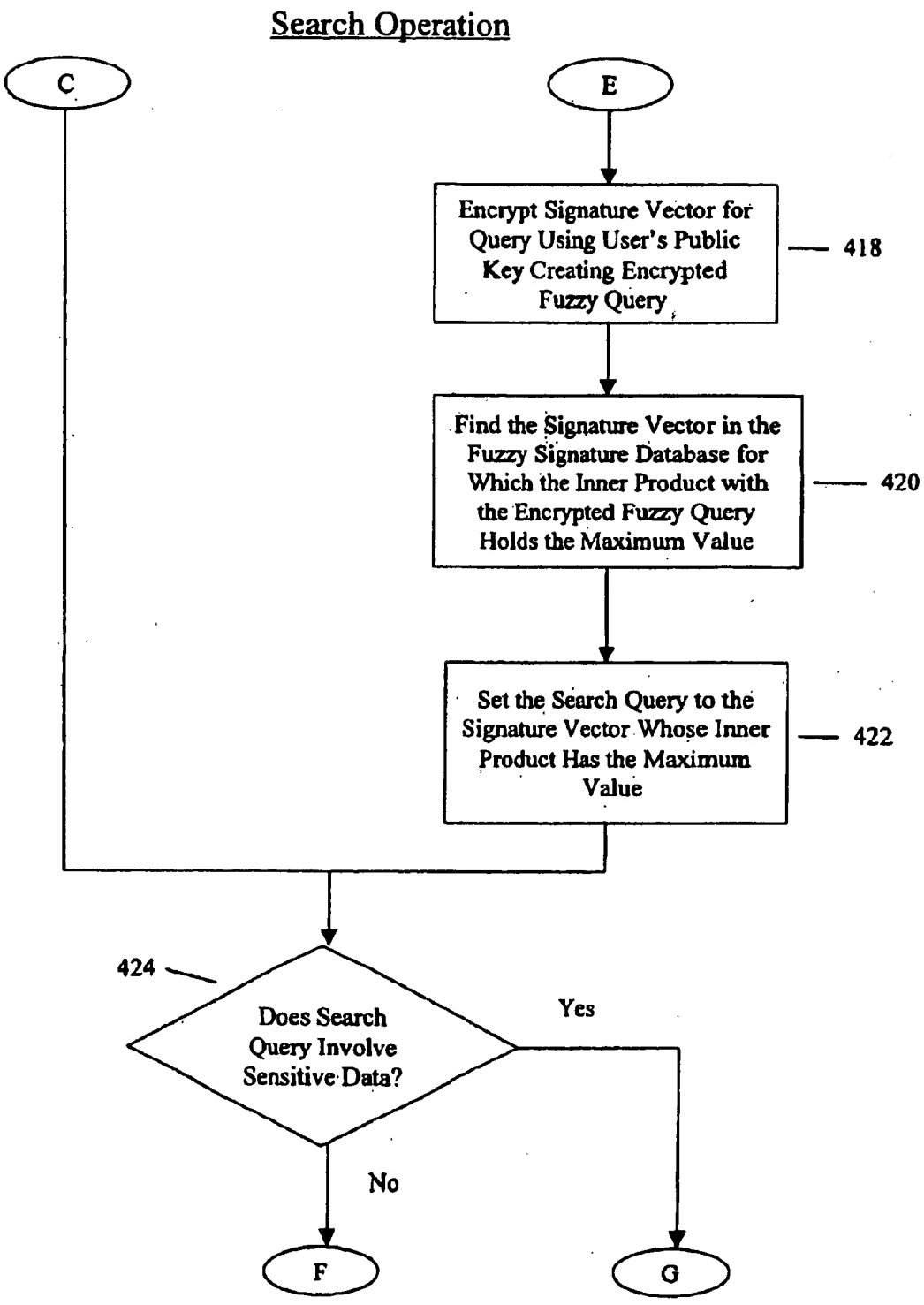


FIG. 4 (continued)

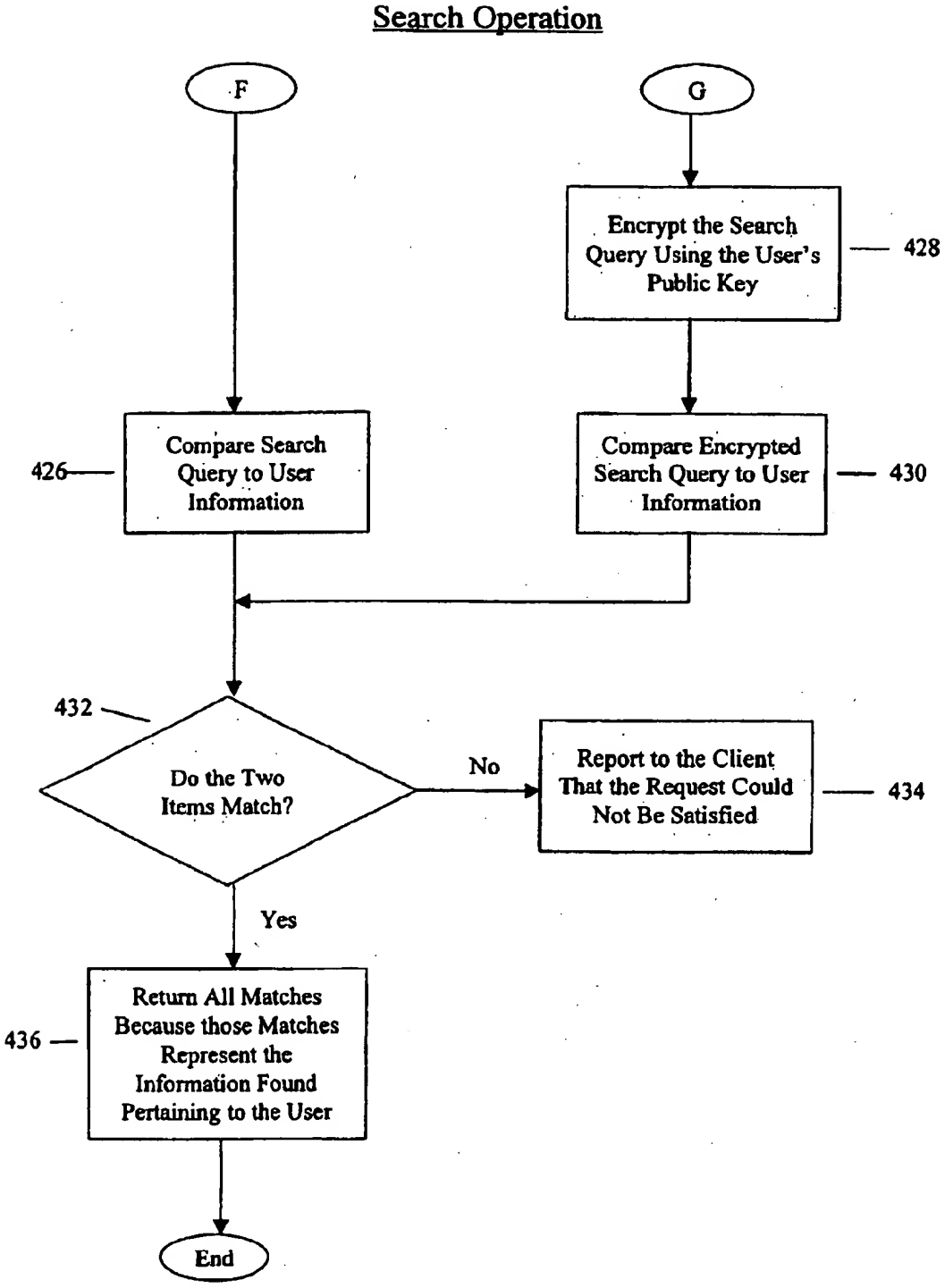


FIG. 5

Accessing Sensitive Data

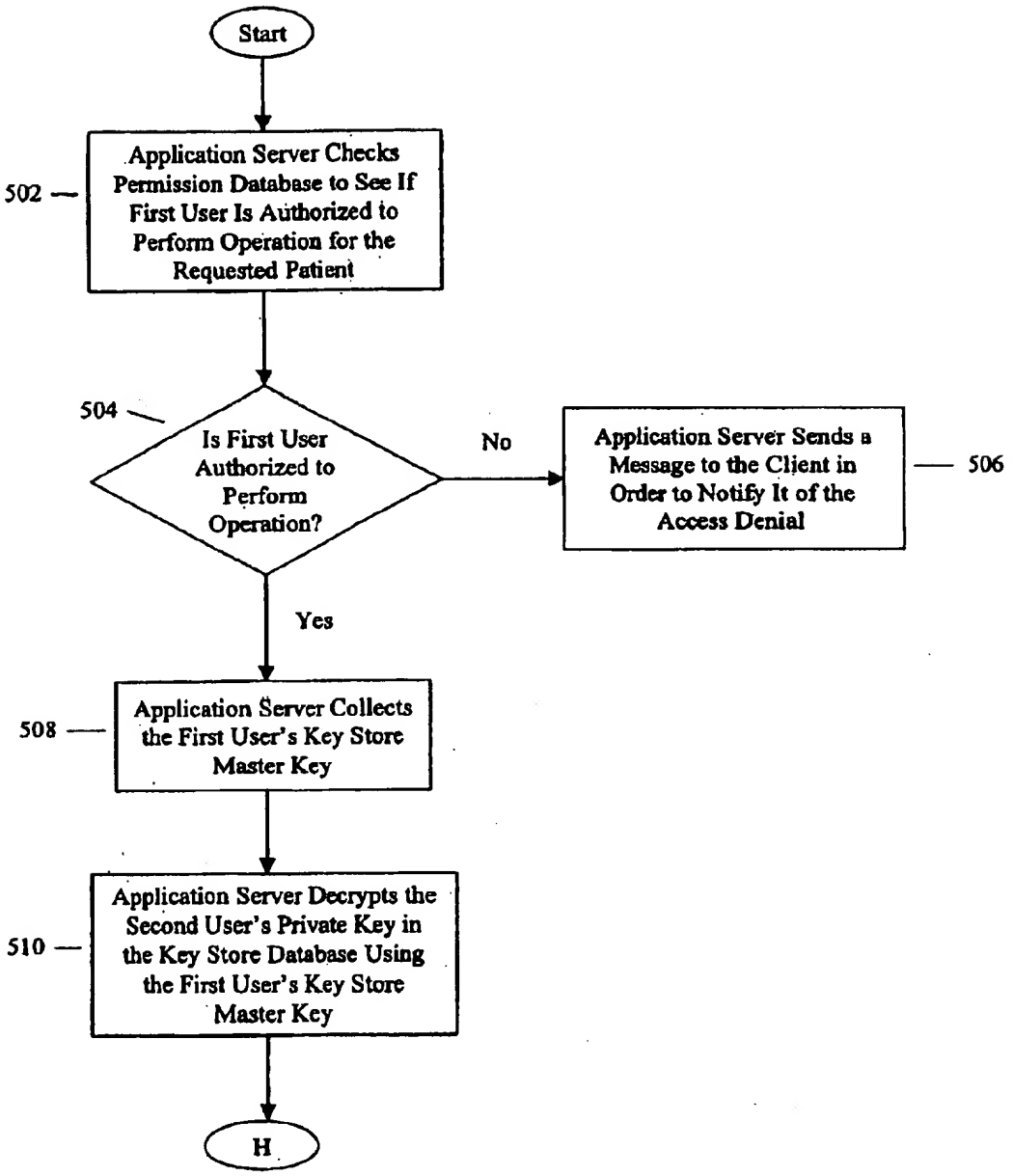
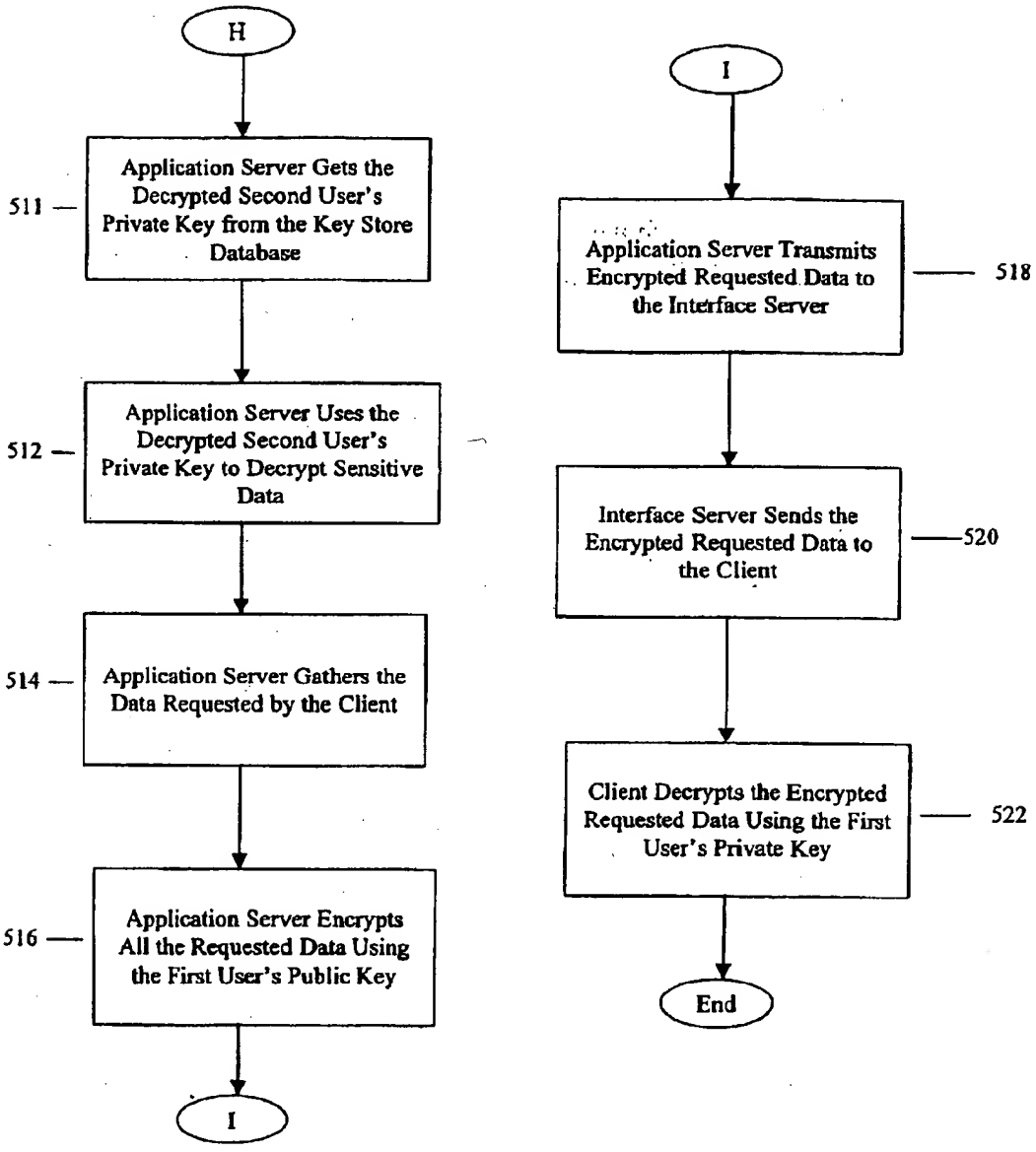


FIG. 5 (continued)

Accessing Sensitive Data (Cont.)



SYSTEM AND METHOD FOR PROVIDING SECURE TRANSMISSION, SEARCH, AND STORAGE OF DATA

FIELD OF THE INVENTION

The present invention relates to computer and network security, and more particularly, a system and method for securely transmitting, searching, and storing data.

BACKGROUND INFORMATION

Advances in computer and communications technology have increased a free flow of information within networked computer systems. While a boon to many, such free flow of information can be disastrous to those systems which process sensitive data. In a typical networked computer system, one or more clients are connected over a communication network to a server.

The risk of a security breach is compounded when a pathway is provided from a private network to a public network such as the Internet. The Internet is a loose conglomeration of networks connected to a standard network protocol. One of the benefits of accessing the Internet is that the vast amounts of information can be accessed by the user. However, of such unobstructed access, the danger is that there are little or virtually no controls on what individuals can access and what they may do with such access. When data is stored or transmitted which allows parties, to access such data even though they are not authorized to access it, it is necessary to take steps to insure the security of that stored data and to ensure the integrity of data transmitted from one computer to another (e.g., via the Internet).

A number of measures, e.g. encryption procedures, have been used to reduce the vulnerability of the networked systems to unauthorized access. Conventional encryption procedures encode data to prevent the unauthorized access, especially during the transmission of the data. Encryption procedure is generally based on one or more keys, or codes, which are essential for decoding, or reverting the data into a readable form.

The traditional encryption techniques focus on the security of the transmission and ignore the security of storage. These techniques provide a protection against the first kind of attacks which include intercepting the data as it is being transmitted. The encryption techniques not only allow the authentication of the sender of a message, but also serve to verify the integrity of the message itself, thus proving that the message has not been altered during the transmission. Such techniques include the use of both symmetric and asymmetric keys, as well as digital signatures and hash algorithms.

The encryption algorithms or procedures are generally characterized in two categories: symmetric and asymmetric. Symmetric algorithms use one key to encrypt and decrypt a message. An encryption key is a sequence of bits that can be used to encode or decode a message. These symmetric algorithms require that both the sender and the intended receiver of the message (and no one else) know the same key. On the other hand, asymmetric algorithms use two separate keys e.g., a public and a private key to encrypt and/or decrypt a message. The public keys are published, (i.e., in the sense that the public key is available from a particular service; such as a telephone directory) so that everyone knows everyone else's public key. The private keys, on the other hand, are kept secret by the owner.

Thus, in a situation where, for example, a patient wanted to send an encrypted message to his or her doctor, the patient would use the doctor's public key to encrypt the message, and then send the encrypted message to the doctor. The doctor would then use his private key to decrypt the message.

The practice of using encryption protocols or procedures to authenticate message senders as well as the integrity of messages is well known in the art (see e.g., Bruce Schneier, Applied Cryptography, Protocols, Algorithms, And Source Code In C, 2d ed., John Wiley & Sons, Inc., 1996).

Conventional systems and methods suffer from, e.g., at least four deficiencies:

1. Restricted media and time: data management security measures only apply to data transmission, thus exposing stored data to an unauthorized access or unauthorized data manipulation;
2. Exceeded user generality: data management security measures ignore interaction patterns between individuals or user groups;
3. Exceeded application scope: security measures ignore specific requirements of particular applications (e.g., medical use); and
4. Exceeded implementation demands: security measures require n−1 keys for a group of n people. (as discussed in the publication by Schneier listed above).

Accordingly, there is a need for a system and method which elevates the security standards across all digital media and prevents compromising data (e.g., patient data) in case of an authorized access of the server. Moreover, there is a need for a system and method that combines security and privacy protection without impeding data processing performance or conventional query scope in a relational database.

SUMMARY OF THE INVENTION

The present invention is directed to a method and system that satisfies the need of securely transmitting, searching, and storing data. Such a system and method allows a user to transfer data securely to a private network by pre-encrypting sensitive data with an encryption key, encrypting both non-sensitive data and the pre-encrypted data with a different encryption key and sending this encrypted data to a private network.

In an embodiment of the system and method, a server is configured to perform fuzzy searching. The procedure for fuzzy searching include creating trigrams for each record in a record database, sorting the trigrams alphabetically, computing signature vectors for each record in the record database, encrypting the signature vectors with an encryption key, and storing the encrypted signature vectors in an encrypted signature database. In addition, the above steps are performed to obtain an encrypted signature vector for a search query. Thereafter, the closest encrypted signature vector is obtained from an encrypted vector database (i.e., the encrypted signature vector that is closest to the search query encrypted signature vector is obtained).

According to another embodiment of the present invention, the record database which contains both non-sensitive data and encrypted sensitive data is searched. This is accomplished by encrypting the search query with an encryption key. Then, one or more records satisfying the search query are found.

Another embodiment of the present invention allows authorized users access to the encrypted sensitive data. First, the database which contains information is checked to determine which users are authorized to access certain data,

and if the user is authorized to access such data, then the user is allowed to access a master encryption key. With the master encryption key, a further encryption key is decrypted. This further encryption key provides access for the user to the sensitive data.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1***a* shows an exemplary embodiment of a system according to the present invention.

FIG. **1***b* shows exemplary keys utilized for authenticating the interface server in the system illustrated in FIG. **1***a*.

FIG. **1***c* shows exemplary keys utilized for transmitting, receiving, and storing data in the system illustrated in FIG. **1***a*.

FIG. **1***d* shows exemplary keys utilized for fuzzy searching a database in the system illustrated in FIG. **1***a*.

FIG. **1***e* shows exemplary keys utilized for relational database searching in the system illustrated in FIG. **1***a*.

FIG. **1***f* shows exemplary keys utilized for accessing sensitive data in the system illustrated in FIG. **1***a*.

FIG. **1***h* shows an exemplary embodiment of a method in which a user communicates with a private network.

FIG. **2** shows another exemplary embodiment of a method according to the present invention for authenticating an interface server.

FIG. **3** shows another exemplary embodiment of a method according to the present invention for securely transmitting and storing data.

FIG. **4** shows another exemplary embodiment of a method according to the present invention for performing the searching operation.

FIG. **5** shows another exemplary embodiment of a method according to the present invention for accessing sensitive data.

DETAILED DESCRIPTION

Overview

The system and method according to the present invention addresses the problems of conventional systems and methods as discussed above by, e.g.: (1) securely transferring and storing sensitive data in a performance enhancing manner by double encrypting preferably only the sensitive data; (2) performing fuzzy searching to allow access to user information knowing a limited amount of information about the user; (3) performing relational database operations on a database that contains unencrypted non-sensitive information and, possibly, encrypted sensitive information; and (4) allowing authorized users access to the sensitive information using a minimum number of keys.

To reduce a performance overhead associated with an information protection process, only a subset of the stored data may be protected (e.g. only a subset of the stored data may be encrypted). Initially, the data is segmented into two basic data types using the sensitivity criteria. For example, in the healthcare industry, the time of the day in which a procedure is scheduled is not sensitive information. Every patient ultimately needs to know when a particular doctor is available, and so, this information should be readily available. However, the name of the patient involved is sensitive information which should be protected.

The system and method is suitable in a situation where there are two interacting groups of users and two classes of records. The two interacting groups of users are a privileged user group and a non-privileged user group. A user can be generally defined as either a privileged user or a non-privileged user. For example, in the healthcare industry, the two interacting groups of users would be the patients (e.g. the non-privileged user group), and the healthcare providers such as doctors (e.g. the privileged user group). The two classes of records would be, e.g.: (1) patient records (e.g., non-privileged user records) and (2) the doctor records (e.g., privileged user records).

For example, the patients grant access privileges (to their records) to some doctors and deny access to all other patients and doctors. The doctors to which privileges were granted, in consultation and agreement with their patients, grant access privileges (to some of those patient records) to some doctors. The doctors also grant access privileges to their research to some other doctors and to some patients (see Table 1 below for a listing of the exemplary privileges which may be granted). The system and method according to the present invention utilizes the asymmetry of the privilege granting scheme to minimize the number of keys used by the participating users.

TABLE 1

Access Privilege Asymmetry

	Patient records	Doctor research records
Patient grants access	To some doctors	No
Doctor grants access	To some doctors and some patients	To some doctors and some patients

Thus, an asymmetric key protocol is used for allowing a transmission of a client's request to a private network. Data is encrypted or decrypted using a public or private key. Algorithms for encrypting and decrypting data are known in the art and include Rivest-Shamir-Adleman encryption ("RSA") and Directory System Agent encryption ("DSA").

FIG. **1***a* shows an exemplary embodiment of the system **10** according to the present invention. An asymmetric key protocol is used for security purposes. A private network (arrangement) **150** includes, e.g., an application server **100**, a record database **105**, a key-store database **115**, a fuzzy signature database **120**, and a permission database **125**.

A public network **135** can be a network in which all users have access without the need for bypassing security measures. An example of the public network **135** is the Internet. An interface server **130** is coupled between the private network **150** and the public network **135** thus allowing users (e.g., physicians and patients) access to information on the private network **150**. A client **140** may include all of the users who require access to information from the private network **150**. When the present invention is used in the medical setting, the client **140** may be a physician or a patient. A verification processor **145** performs a verification function by comparing the two parts of a certificate transmitted by the application server **100** to determine if these two parts match and if so, the interface server **130** is authenticated. The verification processor **145** can be implemented using either hardware or software.

The application server **100** controls the access and retrieval of data between the various databases. The record database **105** is a database containing information such as, e.g., patient records (e.g., patient's name, patient's appointments, disease history, disease diagnosis, etc.) and doctor records (e.g., doctor's research on various diseases).

Some of this information, such as the patient's name and disease history, is sensitive information and thus is encrypted using the patient's public key.

The key-store database **115** is a database containing the users' private keys which are encrypted using a key-store master symmetric key. The fuzzy signature database **120** is a database containing signature vectors for each of the users in the record database **105**, with each signature vector being encrypted using the user's public key. The permission database **125** is a database containing information regarding whether a specific user (such as a doctor or patient) has access to a specific file or record.

The system and method according to the present invention uses an encryption procedure, e.g., at five phases:

1. Selected sensitive information is encrypted at the source level. This information remains encrypted during storage.
2. All data that is to be transmitted is encrypted is encrypted using a key.
3. Private keys are encrypted and stored in the key-store database **115**. All doctors may share a single key to decrypt keys in the key-store database **115**.
4. Sensitive query conditions are encrypted to enable standard SQL searching while preventing retrieval of similar but irrelevant sensitive data.
5. Fuzzy signature vectors for every user record in the record database **105** is encrypted with the user's public key. Such encryption allows fuzzy search for data including specific sub-strings by encrypting the sub-strings with the user's public key and searching the fuzzy signature database **120**.

FIGS. **1***b* to **1***f* shows exemplary keys which can be used for encrypting and decrypting data. In this exemplary embodiment, an asymmetric encryption algorithm may be employed using both public and private keys. Once data is encrypted using a public key, it can only be decrypted using the corresponding private key. Alternatively, if data is encrypted using a private key, it can only be decrypted using the corresponding public key. The public keys can be obtained by anyone from, for example, a service similar to a telephone directory. The private keys, however, are kept secret.

FIG. **1***b* shows the keys which may be used to authenticate the interface server **130**. The application server **100** sends a certificate to the client **140**. The certificate contains both an unencrypted text part and an encrypted part. The encrypted part of the certificate is encrypted using the private key of the verification processor **152**. Once the certificate is received by the client **140**, the encrypted part of the certificate is decrypted using a public key of the verification processor **154**.

FIG. 1c shows the keys which may be used in transmitting, receiving, and storing data. For example, when the client **140** transmits data to the application server **100** in the private network **150**, the client **140** pre-encrypts sensitive data using the user's public key **158** (i.e., the user whose information is to be found from the record database **105**). Then the client **140** encrypts both non-sensitive data and the sensitive data using an application server's public key **160**. After the application server **100** (in the private network **150**) receives the encrypted data from the client **140**, the application server **100** decrypts both the sensitive data and the non-sensitive data using an application server's private key **161**. However, the sensitive data remains secure because that data is only accessible when it is again decrypted using a user's private key **164**.

When the application server **100** transmits data to the client **140**, it uses the user's public key **158** to encrypt the requested data. The user's public key **158**, as defined herein, is the public key of the user, for example in the case of a database search, whose data is to be retrieved from the record database **105**. When the client **140** receives the encrypted data from the application server **100**, it decrypts that data using the user's private key **164**. The user's private key **164**, as defined herein, is the private key of the user, for example in the case of a database search, whose data is to be retrieved from the record database **105**.

FIG. **1***d* shows the key which may be used in a fuzzy search. The application server **100** encrypts all signature vectors corresponding to all records in the record database **105** using the user's public key **158**. In addition, the application server **100** encrypts the signature vector for a particular search query using the user's public key **158**.

FIG. **1***e* shows the key which may be used in a relational database search. If the search query relates to or utilizes sensitive data, then the application server **100** encrypts the search query using the user's (e.g. patient's) public key **158**. Because sensitive information is encrypted in the database which is to be searched, the encryption of the search query allows standard relational database operations to be performed on encrypted data in such a database. Standard relational database operations include searching using, e.g., the SELECT and IF-THEN command.

FIG. **1***f* shows the keys which may be used for accessing sensitive data. The application server **100** uses first user's (e.g., doctor's) key-store master key **168** to decrypt the second user's (e.g., patient's) private key **164** (which is stored as an encrypted key in the key-store database **115**). The application server **100** accesses the sensitive data in the record database **105** by decrypting the sensitive data using the second user's private key **164**. The user's private key **164**, as defined above, is the private key of the user, for example in the case of a database search, whose data is to be retrieved from the record database **105**.

FIG. **1***h* shows exemplary steps of the method according to the present invention for allowing a client **140** to communicate with a private network **150**. In step **20**, the client **140** authenticates the interface server **130** by, e.g., requesting and checking the contents of a certificate. In step **25**, the client **140** determines if the interface server **130** is properly authenticated. If the interface server **130** is not properly authenticated then in step **27**, the client **140** notifies the user of this authentication failure. If the interface server **130** is properly authenticated, then in step **30**, the client **140** transmits data to the application server **100**. In step **40**, the application server **100** performs the operation requested by the client **140** if the client **140** is authorized to perform that operation. If in step **50**, the application server **100** determines that the client **140** is authorized to perform the operation, then in step **60**, the application server **100** returns the requested information to the client **140**. If, however, the application server **100** determines in step **50** that the client **140** is not authorized to perform the requested operation, then in step **70**, it notifies the client **140** of the denial.

Authenticating the Interface Server

FIG. **2** represents the process for authenticating the interface server **130**. In step **200**, the client **140** authenticates the interface server **130** by requesting a certificate from the interface server **130**. The certificate contains two parts, e.g., an encrypted part which is encrypted using the private key of the verification processor **145** and a clear text part (an unencrypted part). The certificate also contains the applica-

tion server's **100** public key. In step **205**, the interface server **130** sends a request for the certificate to the application server **100**. In step **210**, the application server **100** transmits the certificate to the client **140** and encrypts the encrypted portion of the certificate using the private key of the verification processor **145**.

In step **215**, the client **140** (after receiving the certificate from the application server **100**) separates the certificate into two parts, i.e., the encrypted part and the clear text part. In step **220**, the client **140** (using the public key of the verification processor **154** which was sent with the certificate) decrypts the encrypted part of the certificate and in step **225**, determines if the decrypted part matches the clear text part. If both parts match, then in step **235**, the client **140** determines that the interface server **130** is properly authenticated. However, if both parts do not match then, in step **230**, the client **140** displays an error message. This process for authenticating the interface server **130** is can be implemented using, e.g., Verisign in the Microsoft Internet Explorer® or Netscape browsers.

Transmission and Storage of Data

FIG. **3** shows exemplary steps of the exemplary embodiment of the present invention which may be used by the client **180** to transmit a secure request to the application server **100**. In step **305**, the client **140** retrieves the application server's public key **160** from the certificate. In step **310**, the client **140** pre-encrypts sensitive data in the message using the user's (e.g. patient's) public key **158**. In step **315**, the client **140** encrypts all of the message which results in further encrypting the sensitive data with the application server's public key **160**. In step **320**, the client **140** transmits the encrypted message to the interface server **130**. In step **325**, the interface server **130** sends the encrypted message to the application server **100**. In step **330**, the application server **100** decodes the message using its private key **166**. In step **335**, the application server **100** stores the message in the record database **105**. The sensitive data stored in the record database **105** remains encrypted with the user's public key **158**.

The above described method is advantageous because the prior art methods do not perform this double encryption, thus leaving the sensitive information unprotected on the application server **100**. (See e.g., Bruce Schneier, Applied Cryptography, Protocols, Algorithms, And Source Code In C, Pg. 28, 2d ed., John Wiley & Sons, Inc., 1996). With the above described method, the sensitive information remains encrypted, and thus protected on the application server **100**. If a break-in of the private network occurs, the sensitive information remains protected because, e.g., only the user's private key **164** can decrypt that sensitive information.

The Search Operation

In step **40** of FIG. **1***h*, the application server **100** performs the operation requested by the client **140**. This operation can be a search for a particular record, or an insertion or deletion of a record. For a search operation, FIG. **4** shows exemplary steps which may be used to search for a particular record. In step **402**, the application server **100** obtains the search query that the client **140** previously transmitted. In step **404**, the application server **100**, determines if the search requires fuzzy searching. Fuzzy searching is required if the user sitting at the client **140** selects fuzzy searching rather than performing traditional searching which requires an exact match of the search query with a term in the record.

If the fuzzy searching is required, then in step **406**, the application server **100** may create trigrams for every record in the record database **105**. A trigram is a string of three letters. The set of all trigrams for any given portion of text characterizes that text and may be used for its identification in a limited size environment. For example, the word "cryptography" has the following trigrams: "cry", "ryp", "ypt", "pto", "tog", "ogr", "gra", "rap", "aph", and "phy". In step **408**, the trigrams are sorted, e.g., alphabetically. Thus, for the above example, the trigrams would be ordered as: "aph", "cry", "gra", "ogr", "phy", "pto", "rap", "ryp", "tog", and "ypt".

In step **410**, the application server **100** computes a signature vector for each record. The signature vector is a trigram frequency vector for the entire alphabet. In the previous example, the signature vector for the word "cryptography" has 0's in all positions starting with "aaa" and ending with "zzz", except for 1's in the positions of "cry", "ryp", "ypt", "pto", "tog", "ogr", "gra", "rap", "aph", and "phy". For example, the vector for cryptography would have the following values:

aaa	aab	aac	...	aph	...	crp	...	gra	...	ogr	...	phy
0	0	0	...	1	...	1	...	1	...	1	...	1

The signature vector can also be calculated using other methods, such as using quadgrams or pentagrams rather than trigrams.

In step **412**, the application server **100** encrypts the signature vector using the user's public key **158** (i.e., the public key of the user whose information is to be retrieved from the record database **105**). In step **414**, the application server **100** stores the encrypted signature vector in the fuzzy signature database **120**. In step **416**, using, e.g., the above method employing the trigrams, the application server **100** computes the signature vector for the search query. In step **418**, the application server **100** encrypts this signature vector using the user's public key **158** which results in an encrypted fuzzy query.

In step **420**, the application server **100** finds the encrypted signature vector in the fuzzy signature database **120** for which the inner product with the encrypted fuzzy query holds the maximum value. The larger the inner product between the encrypted signature vector and the encrypted fuzzy query, the smaller the cosine of the angle (and thus the smaller the angle) between these two vectors. Computing the inner product is performed using the formula:

$$\sum_{i=1}^{n} x(i) \cdot y(i)$$

where x(i) and y(i) are vectors, and n is the number of dimensions of the vectors. The smaller the angle, the smaller the difference between the vectors which results in finding the signature in the fuzzy signature database that is closest to the query. The system and method according to the present invention, however is not limited to a use of the inner product to find the one vector from a group of vectors that is closest to a query vector. It is also possible to use other conventional methods for finding the one vector from a group of vectors that is closest to the query vector. In step **422**, the search query is set to the signature vector whose inner product has the maximum value.

In step **424**, the application server **100** determines if the search query involves sensitive data (e.g., searching on sensitive data such as the patient's name). If the search query

involves sensitive data then in step **428**, the application server **100** encrypts the search query using the user's public key **158** before searching in the record database **105**. This encryption should be performed because the sensitive information (such as the patient's name) stored in the record database **105** is encrypted with the user's (e.g., patient's) public key **158** (refer to the section above on "Transmission and Storage of Data").

If the search query is not encrypted, then standard relational database operations such as SQL queries would not work when searching for encrypted entries in the database. For example, if a doctor was provided with all the appointments for a particular patient, it is not possible to simply execute a SELECT statement where the patient's name is equal to a certain value because the patient's name (which is sensitive data) is encrypted in the database. Moreover, because the patient's name is encrypted with the patient's public key, it can only be decrypted using the patient's private key. By encrypting the search query, sensitive information can remain encrypted in the database and standard SQL search capabilities can be performed using the encrypted search query. In addition, the patient's private key is not required to perform a search, and therefore sensitive information is not compromised.

In step **430**, the application server **100** compares the encrypted search query to the sensitive patient information in the record database **105**. If the search query does not involve sensitive data, then in step **426**, the application server **100** compares the search query with the user information stored in the record database **105**.

In step **432**, the application server **100** determines if the two particular items match. This determination can be made by, for example, comparing the search query or the encrypted search query with the relevant field of a record in the record database **105**. If the search query requests all patients with the name "John Doe", then the application server **100** searches for all records in the record database **105** whose name field contains the name "John Doe". If the items match, then in step **436**, all the records that match the search query are returned. If the items do not match, then in step **434**, the application server **100** reports to the client **140** that its request could not be satisfied.

After searching and finding the desired information, if that found information contains the sensitive information then that information needs to decrypted using the procedure described below.

Checking Client Authorization

FIG. **5** shows exemplary steps of the exemplary embodiment of the system and method according to the present invention for authorizing the client's request when sensitive information is involved. Such sensitive information resides in the record database **105** and is encrypted with the user's (e.g. patient's) public key **158**. To decrypt that sensitive information and thus be able to use it, the doctor, patient, or another user must be authorized to access that information. If authorized, the application server **100** sends the requested information back to the client **140**. In this embodiment, a first user (e.g., a doctor) is allowed to access sensitive information of second user (e.g., a patient) using only three encryption keys.

In step **502**, the application server **100** determines if a first user is authorized to perform the requested operation by checking the permission database **125**. The permission database **125** contains information as to which users (such as doctors and patients) are allowed to perform operations (e.g., view, search, add, delete, etc.) on the sensitive information located in the records of the record database **105** (e.g., doctor research records or patient records). See Table 1 above for the list of access privileges that doctors and patients can give with regards to patient records and doctor research records.

If the first user is not authorized to perform the requested operation on the sensitive information, then in step **506**, the application server **100** sends a message to the client **140** to notify it of the access denial. If the first user is authorized to perform the requested operation on the sensitive information, then in step **508**, the application server **100** collects the first user's key-store master key **168** which that first user provides. In step **510**, the application server **100**, using the first user's key-store master key **168**, decrypts a second user's private key **164**. While in the key-store database **115**, the second user's private key **164** is encrypted with the key-store master key. In step **511**, the application server **100** obtains the second user's private key from the key-store database **115**. In step **512**, the application server **100** uses the second user's private key to decrypt the sensitive data found in the second user's record in the record database **105**.

The above process of using the key-store master key **168** and the key-store database **115** provided an improvement in that n−1 keys for n people are no longer required. (See e.g., Bruce Schneier, Applied Cryptography, Protocols, Algorithms, And Source Code In C, 2 ed., John Wiley & Sons, Inc., 1996).

By using the key-store master key **168** and the key-store database **115**, only three keys (e.g., the doctor's or first user's public key **158**, the key-store master key **168**, and the patient's or second user's private key **164**) are all that may be necessary to give a doctor access to a patient's records.

The key-store database **115** keeps track of the public and private keys of the users. It enables an authorized user to use another user's private key to decrypt a particular piece of the sensitive data. Usage of the key-store database **115** separates the data from the keys. To prevent an intruder of the application server **100** from gaining access to the users' sensitive data via the patients' private keys stored in the key-store database **115**, all keys stored in the key-store database **115** are encrypted using the key-store master key **168**.

Returning Information to the Client

The application server **100** may send the client requested information, including the sensitive information, back to the client **140**. In step **514**, the application server **100** gathers the data requested by the client **140**. In step **516**, the application server **100** encrypts the requested data using the first user's public key **158**. In step **518**, the application server **100** transmits encrypted requested data to the interface server **130**. In step **520**, the interface server **130** sends the encrypted requested data to the client **140**. In step **522**, the client **140** decrypts the encrypted requested data using the first user's private key. The decrypted sensitive information of the second user is now readable by the first user.

The system and method of the present invention is not limited to the medical industry and in particular where a physician or patient tries to access records of another physician or patient. The system and method may also be applicable in other asymmetric privilege granting environments. For example, the system and method may be used in a corporate environment where an employer has access to employee's records but the employee might have access to only his or her own record, or have access to other employee's records depending on that employee's position in the

company (such as a manager of other employees). The company may have various offices such that a public network would need to be used in order to access certain information from a private network. In this situation, the system and method again capitalizes on the asymmetry of the privilege granting scheme to minimize the number of keys used by the participating users.

Another example is the banking environment where a customer's own bank statement is accessible to that customer but is not accessible to other customers. It is also accessible to certain bank employees such as the loan department or the payment departments. Because of the asymmetry of the privilege granting scheme, this environment can also capitalize on the asymmetry to minimize the number of keys used by the participating users (e.g., bank customers, bank employees, etc.).

While the present invention is described in conjunction with the preferred embodiments, it will be understood that they are not intended to limit the invention to that embodiment. On the contrary, this invention is intended to cover alternatives, modifications, and equivalents, which may be included within the spirit and scope of the invention as defined by the claims. Furthermore, in the previous detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be obvious to one of ordinary skill in the art that the present invention may be practiced without these specific details. In addition, several definitions are provided but it will be appreciated that these definitions are not meant to be limiting but are rather provided for context purposes and that among others, the general definition, as understood by those skilled in the art, also applies. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

What is claimed is:

1. A system for securely transmitting and storing data comprising:

a client arrangement which encrypts sensitive data using a first key to generate pre-encrypted sensitive data, the client encrypting non-sensitive data and the pre-encrypted sensitive data using a second key; and

a private network arrangement including:

a record database including a set of records,

a key-store database including a set of third keys which are encrypted using a fourth key,

a fuzzy signature database including signature vectors which are encrypted using the first key,

a permission database including authorization information, and

an application server locating one of the signature vectors in the fuzzy signature database which substantially corresponds to a query request, performing at least one relational database operation on an encrypted query request, and determining if a first user is authorized to perform an operation, using the authorization information,

wherein, if the first user is authorized, the application server obtains the fourth key for decrypting a particular key of the third keys which corresponds to particular information for a second user stored in the key-store database, and

wherein the application server decrypts the sensitive data obtained from the record database using the particular key.

2. The system of claim **1**, further comprising:

a communication network arrangement connecting the private network arrangement to the client arrangement.

3. The system of claim **2**, wherein the communication network arrangement is a public network arrangement.

4. The system of claim **3**, further comprising:

an interface server connecting the public network arrangement to the private network arrangement.

5. The system of claim **4**, further comprising:

a verification processor authenticating the interface server.

6. The system of claim **1**, wherein the records include non-privileged user records and privileged user records.

7. The system of claim **1**, wherein the first key is a public key of the first user.

8. The system of claim **1**, wherein the second key is a public key of the application server.

9. The system of claim **1**, wherein the third keys include private keys, each key corresponding to a separate user.

10. The system of claim **1**, wherein the fourth key is a key-store master key which is utilized for accessing the key-store database.

* * * * *

US005940507A

United States Patent [19]

Cane et al.

[11] Patent Number: 5,940,507

[45] Date of Patent: Aug. 17, 1999

[54] **SECURE FILE ARCHIVE THROUGH ENCRYPTION KEY MANAGEMENT**

[75] Inventors: **David Cane**, Sudbury; **David Hirschman**, Sharon; **Philip Speare**, Arlington; **Lev Vaitzblit**, Concord, all of Mass.

[73] Assignee: **Connected Corporation**, Framingham, Mass.

[21] Appl. No.: **09/014,830**

[22] Filed: **Jan. 28, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/037,597, Feb. 11, 1997.

[51] **Int. Cl.**[6] **H04L 9/00**

[52] **U.S. Cl.** **380/4**; 380/21; 380/49

[58] **Field of Search** 380/4, 21, 49; 707/204; 711/161, 162; 395/186, 187.01; 713/200, 201

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,235,641	8/1993	Nozawa et al.	380/21
5,416,840	5/1995	Cane et al.	380/4
5,479,654	12/1995	Squibb	395/600
5,559,991	9/1996	Kanfi	395/489
5,584,022	12/1996	Kikuchi et al.	380/21 X
5,719,938	2/1998	Haas et al.	380/21
5,721,777	2/1998	Blaze	380/4
5,748,735	5/1998	Ganesan	380/21
5,764,972	6/1998	Crouse et al.	395/601

OTHER PUBLICATIONS

Bruce Schneier, "Applied Cryptography: Protocols, Algorithms, and Source Code in C", 2nd edition John Wiley and Sons, N.Y. (1995) p. 51 (Key and Message Transmission).

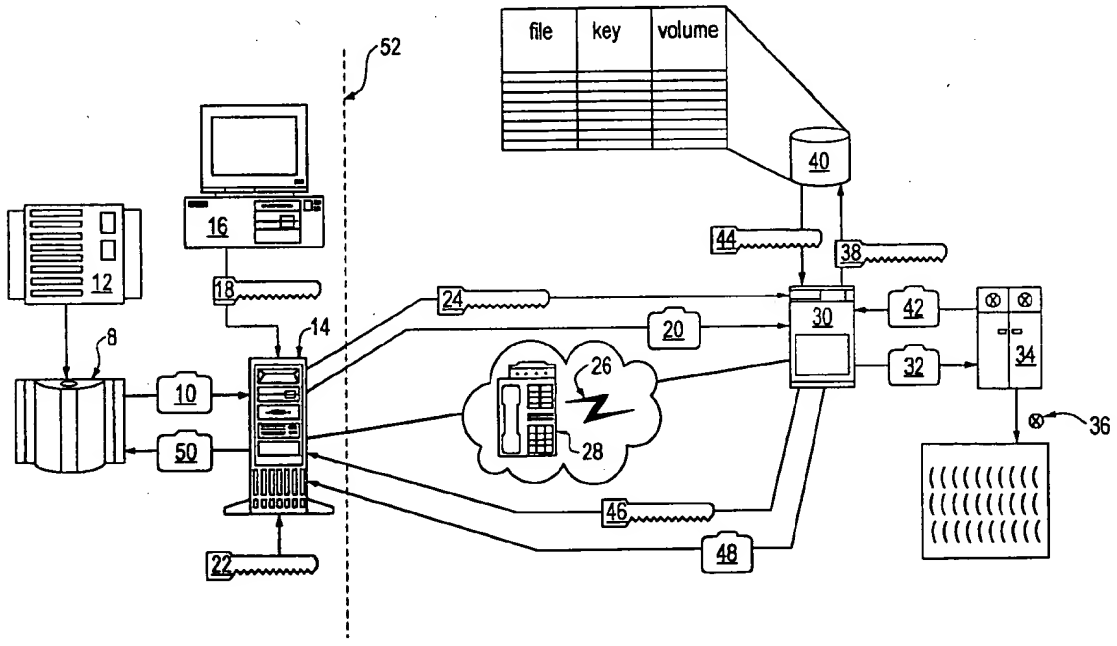
Primary Examiner—Pinchus M. Laufer

Attorney, Agent, or Firm—Weingarten, Schurgin, Gagnebin & Hayes LLP

[57] **ABSTRACT**

A information processing system providing archive/backup support with privacy assurances by encrypting data stored thereby. Data generated on a source system is encrypted, the key used thereby is separately encrypted, and both the encrypted data and encrypted key are transmitted to and maintained by a data repository system. The repository system receives only the encrypted data and key, while the source system retains the ability to recover the key and in turn, the data. The source system is therefore assured of privacy and integrity of the archived data by retaining access control yet is relieved of the physical management of the warehousing medium.

24 Claims, 3 Drawing Sheets



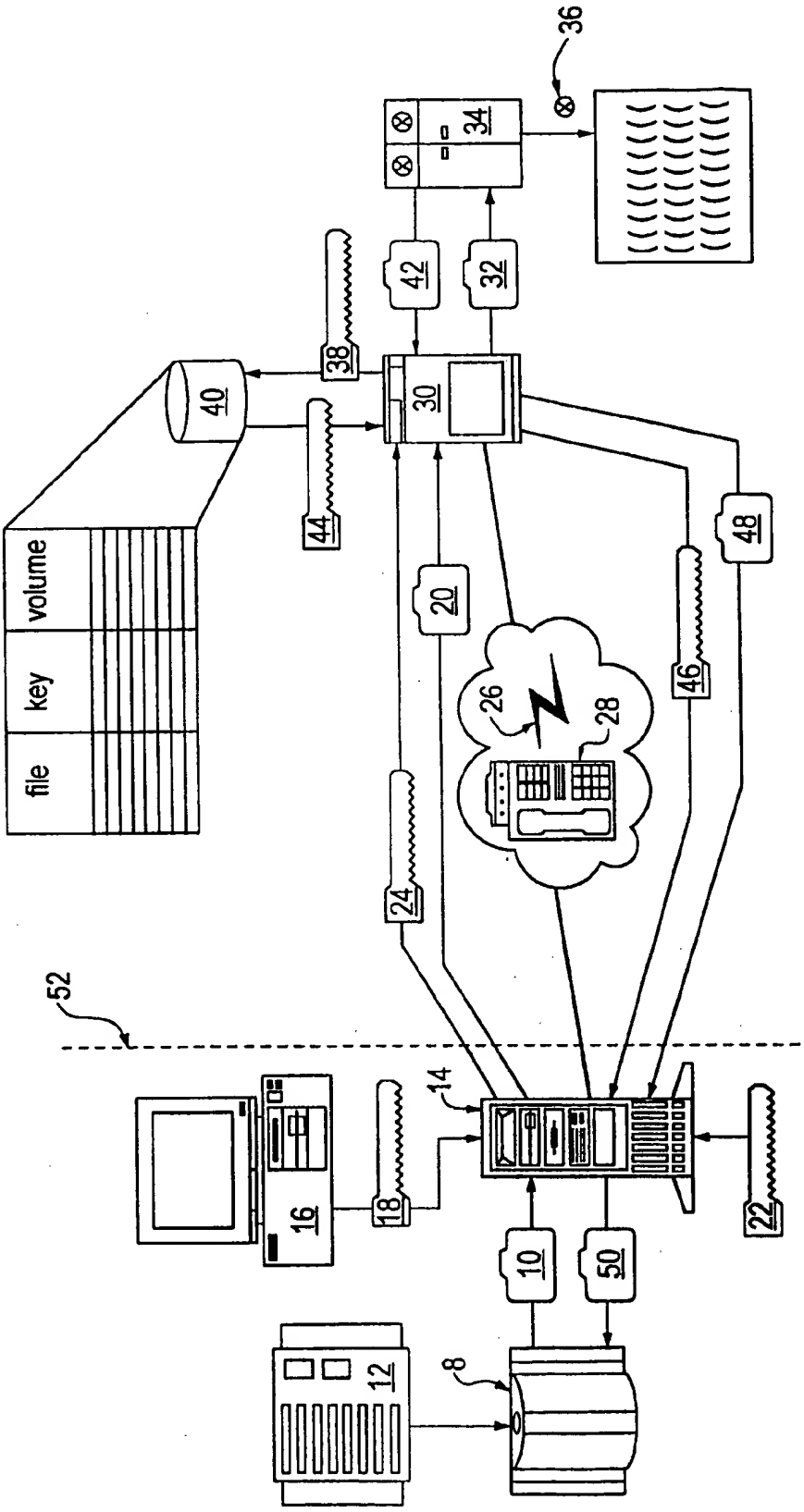

36
34
42
32
38
40
30
44
48
20
volume
key
file
26
28
46
24
52
14
16
18
22
10
50
12
8

FIG. 1

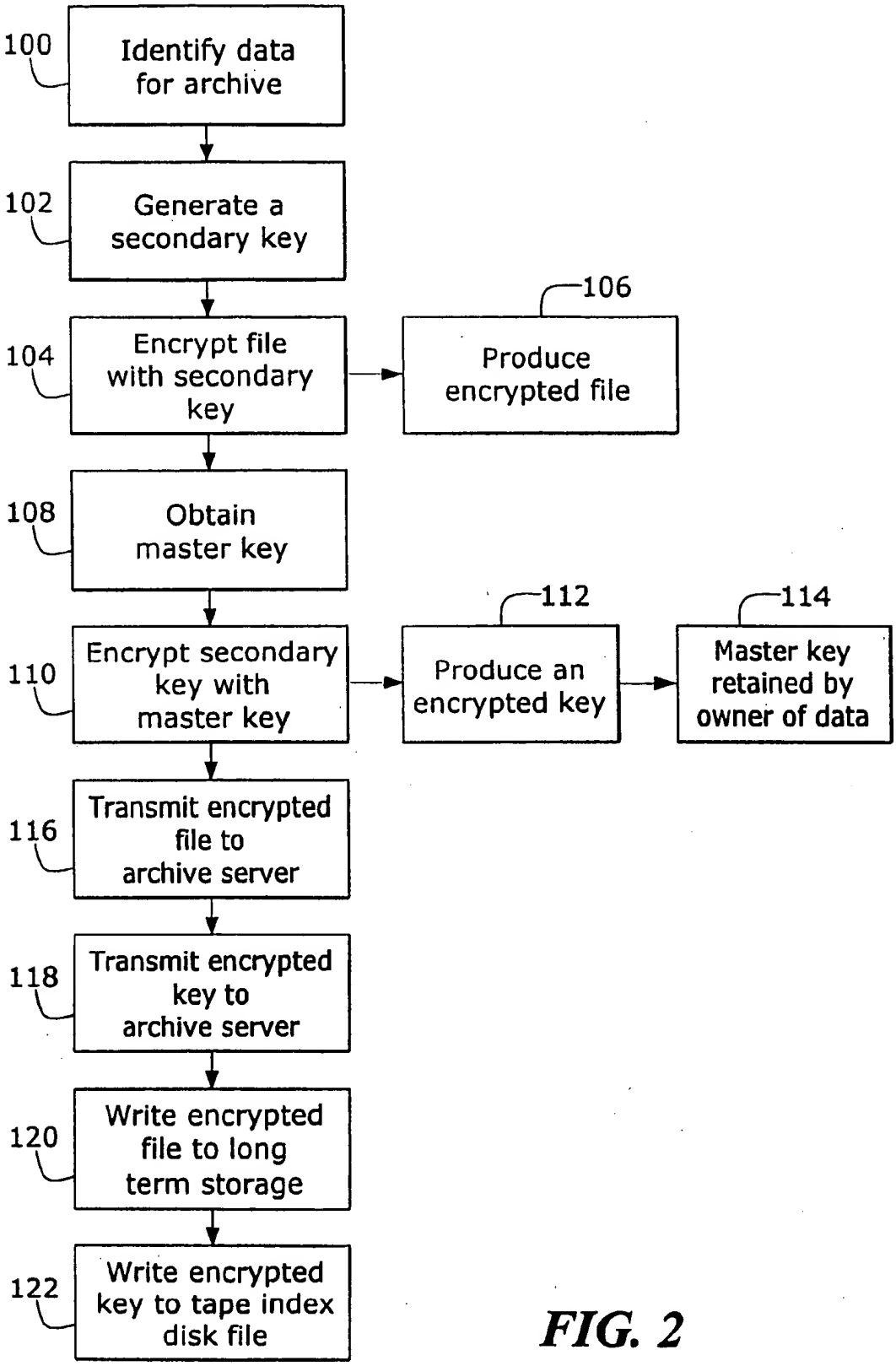

100
Identify data for archive
102
Generate a secondary key
104
Encrypt file with secondary key
106
Produce encrypted file
108
Obtain master key
110
Encrypt secondary key with master key
112
Produce an encrypted key
114
Master key retained by owner of data
116
Transmit encrypted file to archive server
118
Transmit encrypted key to archive server
120
Write encrypted file to long term storage
122
Write encrypted key to tape index disk file

FIG. 2

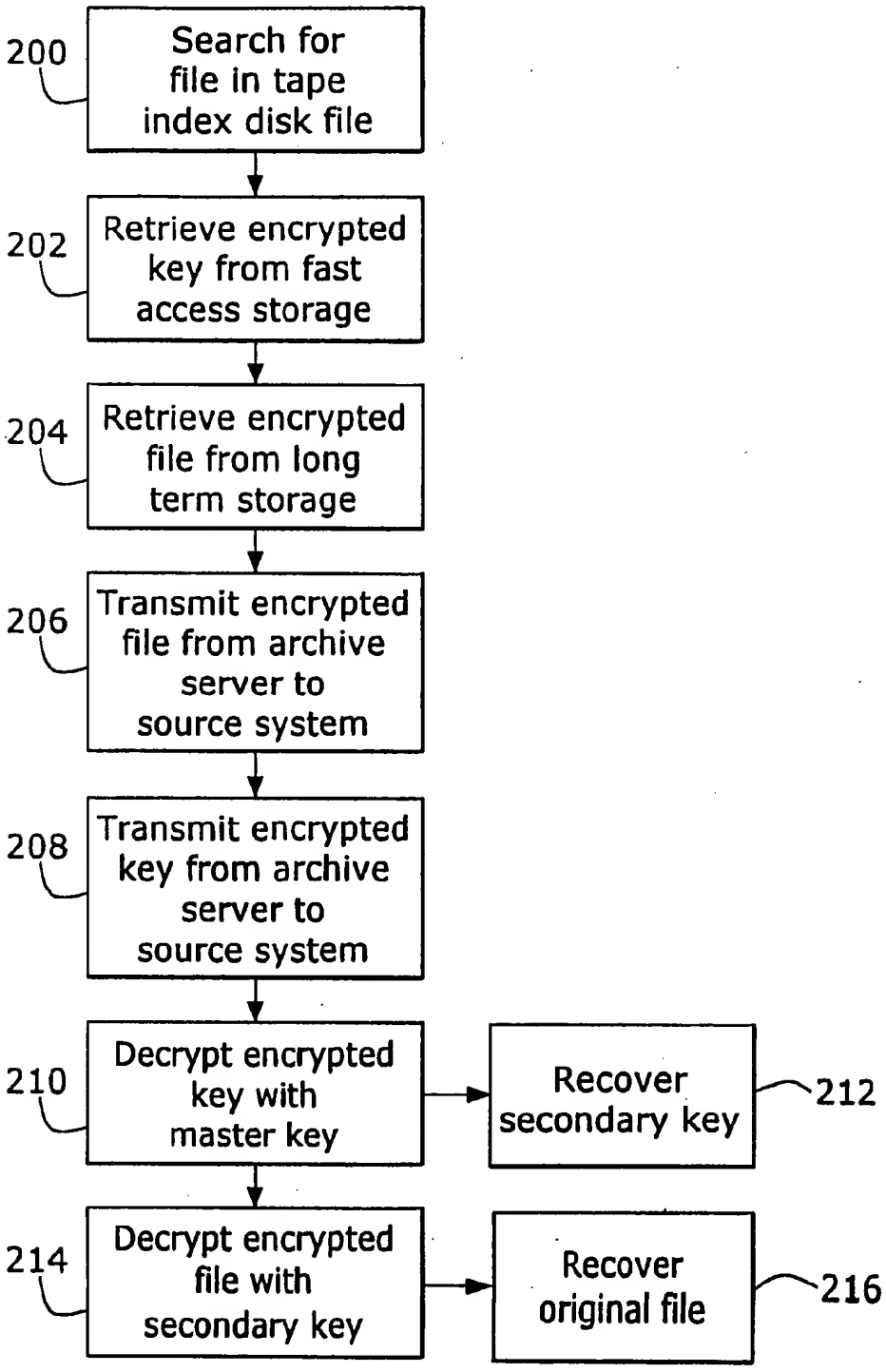

200
Search for file in tape index disk file
202
Retrieve encrypted key from fast access storage
204
Retrieve encrypted file from long term storage
206
Transmit encrypted file from archive server to source system
208
Transmit encrypted key from archive server to source system
210
Decrypt encrypted key with master key
Recover secondary key
212
214
Decrypt encrypted file with secondary key
Recover original file
216

FIG. 3

SECURE FILE ARCHIVE THROUGH ENCRYPTION KEY MANAGEMENT

CROSS REFERENCE TO RELATED APPLICATIONS

A claim of priority is made to U.S. Provisional Patent Application No. 60/037,597, entitled FILE COMPARISON FOR DATA BACKUP AND FILE SYNCHRONIZATION, filed Feb. 11, 1997.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH OR DEVELOPMENT

Not Applicable

FIELD OF THE INVENTION

The present invention relates to data archive operations for information processing systems, and more particularly to security features for such operations.

BACKGROUND OF THE INVENTION

In an information processing system periodic archival of static, unused objects is desirable to optimize access to more active items and to guard against failure such as disk head crashes and human error such as accidental deletions. Consequently, periodic backups to magnetic tape and corresponding purging of selected files from online disks is a common practice.

Data archival mechanisms need to assure the integrity of data stored thereby. Users of the data need to know data is persistent, and also that there is a reasonable turnaround time for retrieval. Often this entails copying such data entities, hereinafter files, to an inexpensive, high volume, but not necessarily fast access, form of physical storage such as magnetic tape. Corresponding index information regarding the magnetic tape location of a particular file can be retained online. Since index information referencing a file consumes much less storage than the file itself, such information is not as unwieldy as the actual data file counterpart. In order to retrieve a file, the index is consulted to determine the physical volume of the corresponding file. The physical magnetic tape volume is then searched for the desired entity. Although sequential, this aspect of the search can be performed within a reasonable time since the indexing system has narrowed the field to a single volume. Such indexing schemes are numerous and are well known to those skilled in the art.

Images written to magnetic tape, however, remain fixed and readable unless physically overwritten. Successive revisions of backups tend to render the previous versions obsolete, although the earlier versions still exist on the tape. Such a tape might well be discarded, thereby placing it in the public domain, or partially used for another purpose, leaving an uncertain status of the information which may exist randomly and unprotected. Further attenuation of control over the data occurs when another party performs the archive. Since the archiving operation usually bears little relation to the generation of the data, it is often desirable to delegate this operation. The archive operation may be undertaken by a co-located group, a group at a remote location of the same organization, or an external contractor, and could involve either electronic or physical mediums of data transmission. Delegation of the backup operation to an archive server, however, raises issues of security and privacy, since the corporation or individual generating the data (hereinafter source organization) has little control over access to the data

at a remote facility. With regard to file deletion, however, magnetic tape does not lend itself well to selective rewrite. Due to the sequential nature of magnetic tape, intra-tape modifications can compromise subsequent files. It is therefore difficult for an archive service to ensure integrity of data upon retrieval requests, provide effective deletion of obsolete data, and maintain secrecy of data while under the control of the archive mechanism.

BRIEF SUMMARY OF THE INVENTION

The present invention addresses the problem of privacy for archived data by providing the source organization with control over the data without burdening the reliability of retrieval with the problems caused by sequential overwrite. An encryption function applied to the archived data renders it in a form unintelligible to unauthorized observers. Encryption involves arithmetic manipulations of the data using a specific value called a key, which renders the data in an unintelligible form. This key bears a specific mathematical relationship to the data and the encryption algorithm being used. Returning the data to the original form involves applying the corresponding inverse function to the encrypted form. Without the proper key, however, it is very difficult to determine the inverse, or decryption, function. The security provided by encryption rests on the premise that with a sufficiently large key, substantial computational resources are required to determine the original data. Encrypting a file with a particular key, and then encrypting the key itself using a master key, therefore, allows another party to physically maintain and store the data while the originator, or source, of the data retains access control. Additional security and authentication measures can also be taken, such as further encrypting the key or the data at the server with a server key, and the use of cipher block chaining to impose dependencies among a sequence of file blocks.

In accordance with the present invention, an archive server utilizes encryption techniques to maintain both security and integrity of stored data by maintaining a series of keys for each archived file, and encrypting both the archived file, and the key to which it corresponds. The archive server manages the encrypted files and the corresponding encrypted keys, while the source organization maintains only the master key required to recover the individual encrypted keys. Through this arrangement, the source organization maintains control and assurances over access to the archived data, while the archive server manages the physical storage medium and performs individual encrypted file manipulation requests at the behest of the client. The archive server maintains access only to the encrypted data files and encrypted keys, effectively managing these files and keys as abstract black-box entities, without the ability to examine and interpret the contents.

Three common transactions involving archived encrypted files are effected by the present invention. A source organization desiring to archive files periodically transfers files from its online repository, usually a fast access storage medium such as a disk, to the archive server. To retrieve archived information, a retrieval transaction indicating a particular file occurs. Finally, when an item is to be deleted, a deletion instruction implicating a particular file is issued to the archive server.

One benefit provided by this arrangement is the elimination of access to data by the archive server, therefore providing the source organization with assurances of access control and privacy, while relieving the source organization of archive cataloging and physical storage duties.

Furthermore, effective deletion of information stored on archive tapes is achieved without physical modification to magnetic tape, therefore avoiding compromise to subsequent data on the same volume.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWING

The invention will be more fully understood in view of the following Detailed Description of the Invention and Drawing, of which:

FIG. **1** is a block diagram of the physical information flow;

FIG. **2** is a flowchart depicting the archival method; and

FIG. **3** is a flowchart depicting the retrieval method.

DETAILED DESCRIPTION OF THE INVENTION

U.S. Provisional Patent Application No. 60/037,597 entitled FILE COMPARISON FOR DATA BACKUP AND FILE SYNCHRONIZATION, filed Feb. 11, 1997, is incorporated herein by reference.

Referring to FIG. **1**, in a computer information processing system large amounts of data are stored and must periodically be archived. Often data is copied from a source system **8** to an archive information processing system **30**, hereinafter archive server, over a transmission medium, **26** & **28**. The archive server **30** then copies the data to be archived onto a suitable long term storage volume such as magnetic tape **36**.

An archive transaction for a file stored at the source system encompasses encryption of the file on the source system using a secondary key, encryption of the secondary key on the source system using a master key, and transmission of the encrypted file and the associated encrypted key to the archive server. Transmission is electronic via computer network, or in alternative embodiments by physical delivery of a suitable magnetic medium. The archive server then stores the encrypted file on magnetic tape or another medium of long term storage, and stores the encrypted key along with an index to the tape containing the encrypted file. The master key used to encrypt the secondary key is retained on the source system.

Referring to FIGS. **1** and **2**, A file **10** to be archived is identified **100** within a fast access storage medium **12** of the source information system **8**, and is sent to a cryptographic engine **14**. The present embodiment incorporates a disk drive as the fast access storage medium, although an alternative embodiment could use other modes of digital fixation, such as CD-ROM. The cryptographic engine **14** may be an application within the same node or an independent CPU, and may invoke specialized encryption hardware, depending on the encryption method desired. Any of various known encryption methods could be employed.

A key generator **16** then generates a secondary key **18** as shown in step **102**, and uses this key to encrypt the file **10** as shown in step **104** to produce an encrypted file **20**, at step **106**. The master encryption key **22** is then obtained in step **108** and used to encrypt the secondary key in **18**, as shown at step **110**, and produce an encrypted key **24**, as indicated in step **112**. Note that since the same master key is used to encrypt multiple secondary keys it need be generated only once and then reused for successive secondary keys. The encrypted file **20** and encrypted key **24** are then transmitted to the archive server at steps **116** and **118**, respectively, while the master key **22** is retained at the source system **8** at step **114**. Transmission may be accomplished via Internet **26**, dialup connection **28**, or in alternative embodiments, other means such as physical delivery of the storage medium. Encryption may be performed by any of various known methods, such as RSA, DES, and other permutations and may involve authentication and verification either through a trusted third party or mathematical methods. Such authentication and verification may involve cipher block chaining (CBC), to perform an XOR on all or part of a previous block and use the resultant value in encrypting a successive block, or checksums such as cyclic redundancy checks (CRC), MD4, and MD5, which accumulate all values in a particular block according to a mathematical formula to arrive at a value which is highly unlikely to be duplicated if data in the block is changed or lost.

Upon receipt of the encrypted file **20** and the encrypted key **24**, the archive server **30** writes the encrypted file **32** to a magnetic tape **36**, or other medium of long term storage which is inexpensive and which need not encompass real time access, via tape drive **34** at step **120**. The encrypted key **38** is then written to a tape index disk file **40** at step **122**, thereby associating the magnetic tape volume **36** with the encrypted file **32** and the encrypted key **38**. In alternative embodiments, a further encryption operation may be performed at the archive server on the encrypted file **32** or the encrypted key **38** to add an additional layer of security.

Recovery of a file is accomplished by the archive server referencing the index to obtain the encrypted key and the volume of the encrypted file. The encrypted file is then retrieved from the volume, and both the encrypted file and encrypted key are transmitted back to the client. The client then recovers the file through the same two stage process used to encrypt. First, the secondary key must be recovered by decrypting the encrypted key with the master. Second, the original file may be recovered by decrypting the encrypted file with the secondary key.

Referring to FIGS. **1** and **3**, for file recovery the archive server searches the tape index disk file **40** at step **200** to lookup the encrypted key **44** and the location of the magnetic tape volume **36**. The server then retrieves the encrypted key at step **202** and retrieves the encrypted file **42** from long term storage via tape drive **34**, as shown in step **204**. The encrypted file **48** and encrypted key **46** are then transmitted back to the source system **8** as indicated by steps **206** and **208**, respectively.

Once received by the source system **8**, the master key **22** is used to decrypt the encrypted key **46** at step **210** and recover the secondary key **18**, as shown in step **212**. The secondary key **18** is then used to decrypt the encrypted file **48** as shown in step **214** to produce the recovered file **50** which is identical to the original file **10**, as indicated by step **216**.

File deletion involves searching the tape index disk file **40**, for the entry corresponding to the file **10** marked for deletion. Rather than retrieving the key and volume, however, the encrypted key **44** is deleted and the storage area in the tape index disk file **40** overwritten with zero values. This overwriting is required to avoid future access to the encrypted key **44** through use of a sector level disk access, as many file systems merely flag a deleted area as available, and data physically remains unaltered until a subsequent write needs the available space. Elimination of the encrypted key effectively precludes future access to the contents of the archived file stored on magnetic tape without requiring physical modification to the archive volume; only the encrypted key is deleted. Therefore, there is no compro-

mise of the integrity of adjacent entities on the tape, and no extraneous versions of sensitive data.

Following overwrite of the encrypted key **44**, the information in the encrypted file **32** remains secure. No modification of the magnetic tape volume **36** is required, as the encryption ensures that the information remains unintelligible.

Effectiveness of this method suggests that the encryption take place no more remotely than the limits of the source system organization's proprietary, or internal, network, as unprotected electronic transfers can also compromise the data. The dotted line **52** on FIG. **1** indicates the extent of unencrypted data and should represent no greater extent than the intranet of the originating entity.

Master key generation is significant because recovery of a key allows recovery of the file that the key represents. Consequently, control over access and deletion to archived files is dependent upon control over the corresponding secondary keys. Each key, however, must be unique to the file to which it corresponds, otherwise, exposure of a key to decrypt a particular file compromises that key for all other files which that key covers. If the source system is required to maintain a separate key for all archived encrypted files, however, there is merely a shift in storage medium, as the key to each encrypted file, rather than the file, must be still be maintained. Encrypting individual secondary keys allows the keys to be maintained as securely as the files. The source system maintains a single master key, or several master keys covering different groups of secondary keys. Control of the archived, encrypted files is then focused through a master key. The archiving entity retains a set of all encrypted files, and maintains a mapping to the corresponding encrypted keys for which the source organization holds the master key.

Having described the preferred embodiments of the invention, other embodiments which incorporate concepts of the invention will now become apparent to one skilled in the art. Therefore, the invention should not be viewed as limited to the disclosed embodiments but rather should be viewed as limited only by the spirit and scope of the appended claims.

What is claimed is:

1. An electronic network for transferring data units among storage elements comprising:

a communications link;

a source information processing system at a first end of said communications link further comprising:

a master encryption key;

at least one secondary encryption key;

a first memory for storing data units and said master and said at least one secondary encryption keys; and

an encryption engine for selectively encrypting said data units to produce encrypted data units using at least one of said secondary encryption keys, and for encrypting said at least one secondary encryption key with said master encryption key producing at least one encrypted key; and

an archive server information processing system having at least one archive server key at a second end of said communications link comprising a second memory and in communication with said source information processing system, said archive server information processing system for receiving and storing said encrypted data units and said encrypted keys in said second memory wherein said archive server key is used to further encrypt said encrypted keys.

2. The network as in claim **1** wherein said first and said second memories provide fixation in a medium selected from the group consisting of electronic, magnetic, and optical storage media.

3. The network as in claim **1** wherein said first memory comprises a substantially real-time random access storage medium.

4. The network as in claim **1** wherein said second memory comprises a first and second storage area, said first storage area comprising substantially real-time random access storage medium, and said second storage area comprising high-volume storage wherein storage capacity and speed are not degraded by quantity of information stored thereby.

5. The network as in claim **4** wherein said high-volume storage is comprised of detachable physical volumes capable of selective and repeatable communication with said archive server information processing system.

6. The network as in claim **4** wherein said at least one encrypted key is stored in said first storage area within said second memory and said encrypted data units are stored in said second storage area within said second memory.

7. The network as in claim **1** wherein said data units comprise elements of a file system.

8. The network as in claim **1** wherein said data units comprise a discrete and enumerable area within said first memory.

9. The network as in claim **1** wherein said source information processing system further comprises a computer and said encryption engine is implemented by said computer executing an encryption application having said master encryption key, said at least one secondary key, and said data units as inputs and said encrypted data units and said at least one encrypted key as outputs.

10. The network as in claim **1** wherein said source information processing system further comprises a computer and said encryption engine is implemented by a circuit in communication with said computer, said circuit having said master encryption key, said at least one secondary encryption key, and said data units as inputs and said encrypted data units and said at least one encrypted key as outputs.

11. The network as in claim **1** further comprising a plurality of said source information processing systems electrically connected to said archive server information processing system.

12. The network as in claim **1** wherein said data units comprise subdivisions comprising a plurality of blocks and said encryption is applied to said blocks wherein input to said encryption includes values from said plurality of blocks and the results of at least one previous encrypted block.

13. An electronic network for transferring data units among storage elements comprising:

a communications link;

a source information processing system at a first end of said communications link further comprising:

a master encryption key;

at least one secondary encryption key;

a first memory for storing data units and said master and said at least one secondary encryption keys; and

an encryption engine for selectively encrypting said data units to produce encrypted data units using at least one of said secondary encryption keys, and for encrypting said at least one secondary encryption key with said master encryption key producing at least one encrypted key; and

an archive server information processing system having at least one archive server key at a second end of said communications link comprising a second memory and in communication with said source information processing system, said archive server information pro-

cessing system for receiving and storing said encrypted data units and said encrypted keys in said second memory wherein said archive server key is used to further encrypt said encrypted data units.

14. A method for providing secure archive for data generated in a first memory within a source information processing system comprising the steps of:

identifying data for archive within said first memory;

obtaining a secondary encryption key;

encrypting said data with said secondary encryption key to produce encrypted data;

obtaining a master encryption key;

encrypting said secondary encryption key with said master encryption key to produce an encrypted key;

transmitting said encrypted data and encrypted key to an archive information system having a second memory;

writing said encrypted data and said encrypted key to said second memory; and

overwriting the portion of said second memory where said encrypted key is stored.

15. The method according to claim 14 wherein the step of transmitting comprises sending via electromagnetic medium.

16. The method according to claim 14 wherein the step of transmitting is selected from the group consisting of transmitting via electronic network communications and transmitting via dedicated telephone modem connection.

17. The method according to claim 14 wherein the step of identifying data for archive is comprised of demarcating an enumerated area within said first memory.

18. The method according to claim 14 wherein the step of identifying data in first memory comprises locating information from fixation in a medium selected from the group consisting of magnetic, electronic and optical.

19. The method according to claim 14 wherein the step of writing to second memory consists of fixation in a medium selected from the group consisting of magnetic, electronic and optical.

20. The method according to claim 14 wherein said data is subdivided into a plurality of blocks and input to said encrypting includes the results of at least one previous encrypting of said blocks.

21. A method for providing secure archive for data generated in a first memory within a source information processing system comprising the steps of:

identifying data for archive within said first memory;

obtaining a secondary encryption key;

encrypting said data with said secondary encryption key to produce encrypted data;

obtaining a master encryption key;

encrypting said secondary encryption key with said master encryption key to produce an encrypted key;

transmitting said encrypted data and encrypted key to an archive information system having a second memory and an archive server encryption key;

further encrypting said encrypted key with said archive server encryption key;

writing said encrypted data and said encrypted key to said second memory.

22. A method for providing secure archive for data generated in a first memory within a source information processing system comprising the steps of:

identifying data for archive within said first memory;

obtaining a secondary encryption key;

encrypting said data with said secondary encryption key to produce encrypted data;

obtaining a master encryption key;

encrypting said secondary encryption key with said master encryption key to produce an encrypted key;

transmitting said encrypted data and encrypted key to an archive information system having a second memory and an archive server encryption key;

further encrypting said encrypted data with said archive server encryption key;

writing said encrypted data and said encrypted key to said second memory.

23. A method for providing secure archive for data generated in a first memory within a source information processing system comprising the steps of:

identifying data for archive within said first memory;

obtaining a secondary encryption key;

encrypting said data with said secondary encryption key to produce encrypted data;

obtaining a master encryption key;

encrypting said secondary encryption key with said master encryption key to produce an encrypted key;

transmitting said encrypted data and encrypted key to an archive information system having a second memory and an archive server encryption key;

writing said encrypted data and said encrypted key to said second memory

retrieving said encrypted data and said encrypted key from said second memory of said archive information system;

decrypting said encrypted key with said archive server encryption key;

transmitting said encrypted data and said encrypted key from said archive information system to said source information processing system;

decrypting said encrypted key with said master encryption key to recover said secondary key; and

decrypting said encrypted data with said secondary key to recover said data.

24. A method for providing secure archive for data generated in a first memory within a source information processing system comprising the steps of:

identifying data for archive within said first memory;

obtaining a secondary encryption key;

encrypting said data with said secondary encryption key to produce encrypted data;

obtaining a master encryption key;

encrypting said secondary encryption key with said master encryption key to produce an encrypted key;

transmitting said encrypted data and encrypted key to an archive information system having a second memory and an archive server encryption key;

writing said encrypted data and said encrypted key to said second memory;

retrieving said encrypted data and said encrypted key from said second memory of said archive information system;

decrypting said encrypted data with said archive server encryption key;

transmitting said encrypted data and said encrypted key from said archive information system to said source information processing system;

decrypting said encrypted key with said master encryption key to recover said secondary key; and

decrypting said encrypted data with said secondary key to recover said data.

* * * * *

US005606315A

United States Patent [19]

Gaskins

[11] Patent Number: 5,606,315

[45] Date of Patent: Feb. 25, 1997

[54] **SECURITY METHOD FOR PROTECTING ELECTRONICALLY STORED DATA**

[75] Inventor: **Ronald E. Gaskins**, Kokomo, Ind.

[73] Assignee: **Delco Electronics Corp.**, Kokomo, Ind.

[21] Appl. No.: **353,745**

[22] Filed: **Dec. 12, 1994**

[51] **Int. Cl.**[6] **G01M 15/00**

[52] **U.S. Cl.** **340/825.34**; 73/116; 324/378; 340/825.31; 380/4; 380/23; 364/424.04

[58] **Field of Search** 73/116, 117.3; 324/378; 340/825.31, 825.34; 380/4, 23, 25; 364/424.03, 424.04, 431.01

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,271,482	6/1981	Giraud	340/825.34
4,546,646	10/1985	Takahashi	73/117.3
4,588,991	5/1986	Atalla	340/825.34
4,677,429	6/1987	Glotzbach	364/424.04
4,800,590	1/1989	Vaughan	380/23
4,926,330	5/1990	Abe et al.	364/431.01
4,959,860	9/1990	Watters et al.	380/25 X
4,996,643	2/1991	Sakamoto et al.	364/431.01
5,003,479	3/1991	Kobayashi et al.	364/431.01
5,060,263	10/1991	Bosen et al.	380/25
5,115,508	5/1992	Hatta	340/825.34
5,222,135	6/1993	Hardy et al.	380/4
5,226,080	6/1993	Cole et al.	380/25
5,365,587	11/1994	Campbell et al.	340/825.31 X
5,402,492	3/1995	Goodman et al.	380/25
5,475,762	12/1995	Morisawa et al.	380/25

FOREIGN PATENT DOCUMENTS

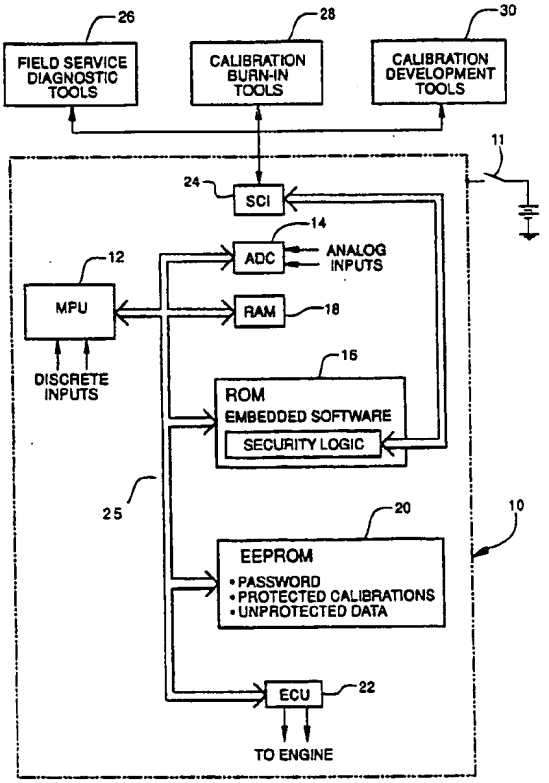
647393	1/1991	Australia	364/424.03
310138	12/1990	Japan	364/424.04
6-160245	6/1994	Japan	364/424.03
9215852	9/1992	WIPO	73/117.3

Primary Examiner—Brian Zimmerman
Assistant Examiner—William H. Wilson, Jr.
Attorney, Agent, or Firm—Mark A. Navarre

[57] **ABSTRACT**

A microprocessor based electronic control module with an EEPROM for storing protected data allows the data to be used internally, and allows non-sensitive data to be accessed by external communication tools, but prohibits access to the protected data unless a password is first entered. Then the data may be read from memory and the data or the password may then be changed. For a given model of control module, an ID number is assigned to the password and stored in the module, and can be read to allow the user to find the corresponding password on a secure list available only to authorized personnel. When a password can not be found and it is necessary to change the protected data, the unit can be recovered by a recover procedure wherein the secure data is first erased and then the security is deactivated to grant free access.

9 Claims, 3 Drawing Sheets



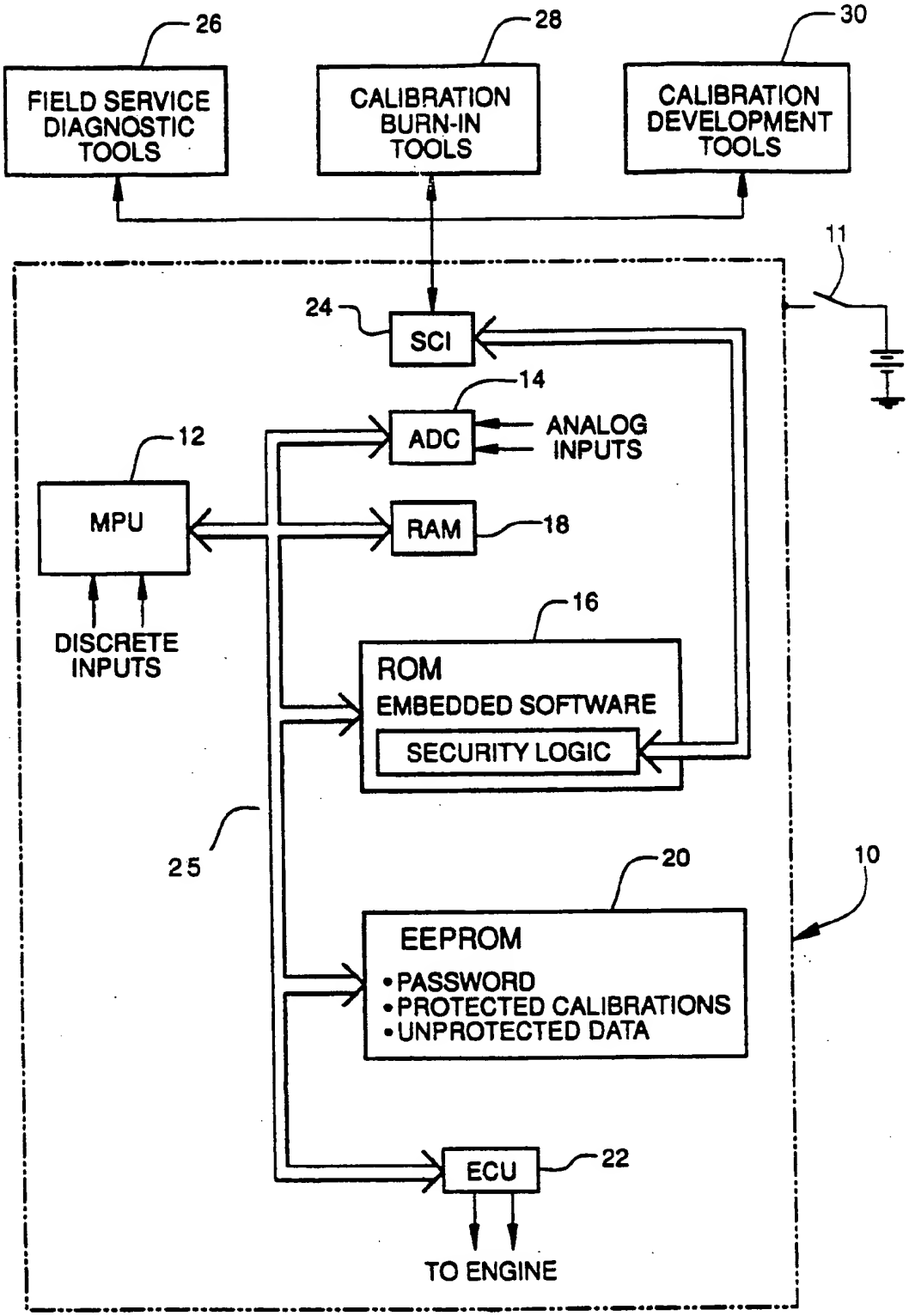
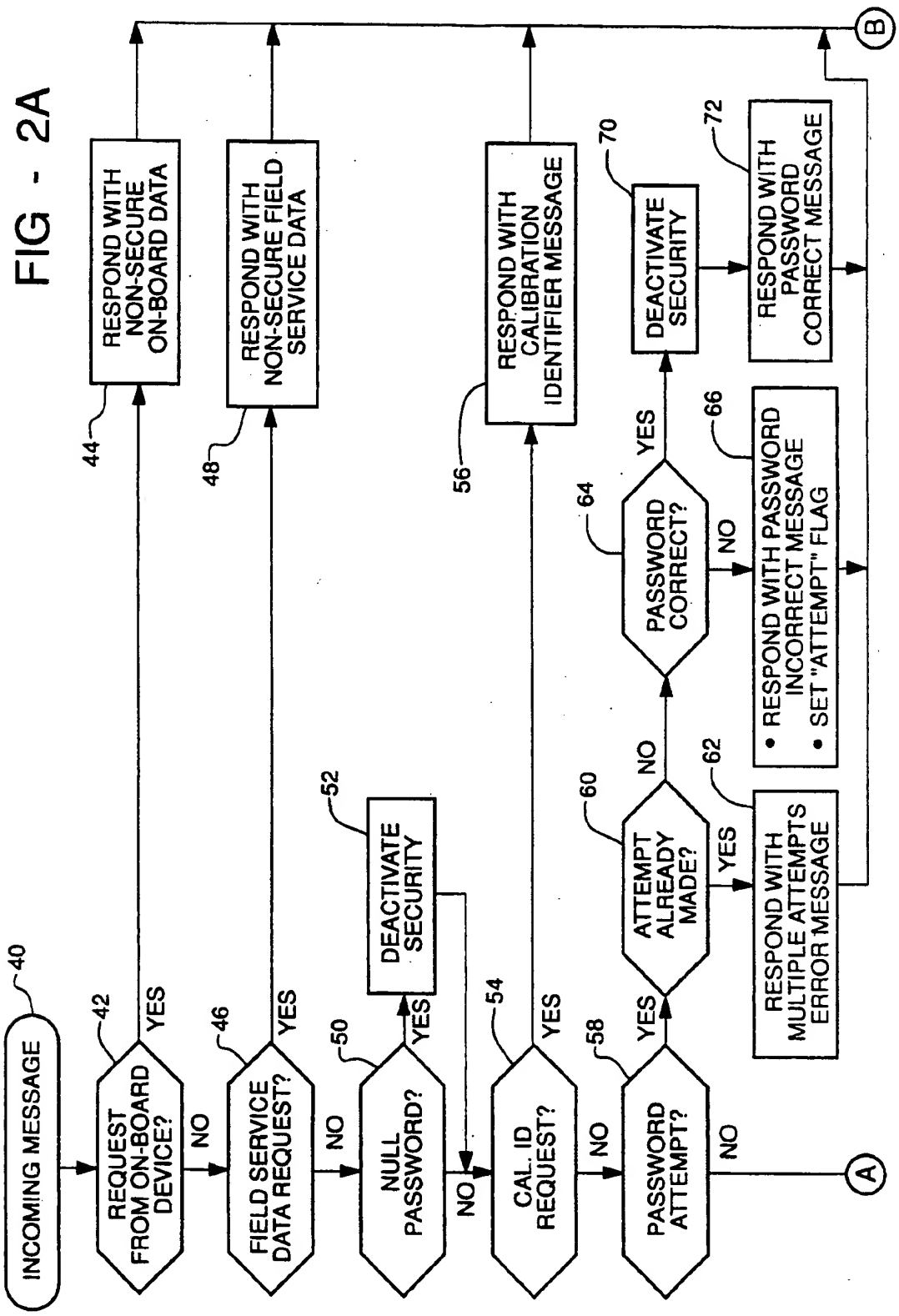

26
28
30
FIELD SERVICE DIAGNOSTIC TOOLS
CALIBRATION BURN-IN TOOLS
CALIBRATION DEVELOPMENT TOOLS
11
24
SCI
14
ADC
ANALOG INPUTS
12
MPU
RAM
18
DISCRETE INPUTS
16
ROM
EMBEDDED SOFTWARE
SECURITY LOGIC
25
20
10
EEPROM
•PASSWORD
•PROTECTED CALIBRATIONS
•UNPROTECTED DATA
ECU
22
TO ENGINE

FIG - 1

FIG - 2A


INCOMING MESSAGE
40
REQUEST FROM ON-BOARD DEVICE?
42
YES
NO
RESPOND WITH NON-SECURE ON-BOARD DATA
44
FIELD SERVICE DATA REQUEST?
46
YES
NO
RESPOND WITH NON-SECURE FIELD SERVICE DATA
48
NULL PASSWORD?
50
YES
NO
DEACTIVATE SECURITY
52
CAL. ID REQUEST?
54
YES
NO
RESPOND WITH CALIBRATION IDENTIFIER MESSAGE
56
PASSWORD ATTEMPT?
58
YES
NO
A
ATTEMPT ALREADY MADE?
60
YES
NO
RESPOND WITH MULTIPLE ATTEMPTS ERROR MESSAGE
62
PASSWORD CORRECT?
64
YES
NO
• RESPOND WITH PASSWORD INCORRECT MESSAGE
• SET "ATTEMPT" FLAG
66
DEACTIVATE SECURITY
70
RESPOND WITH PASSWORD CORRECT MESSAGE
72
B

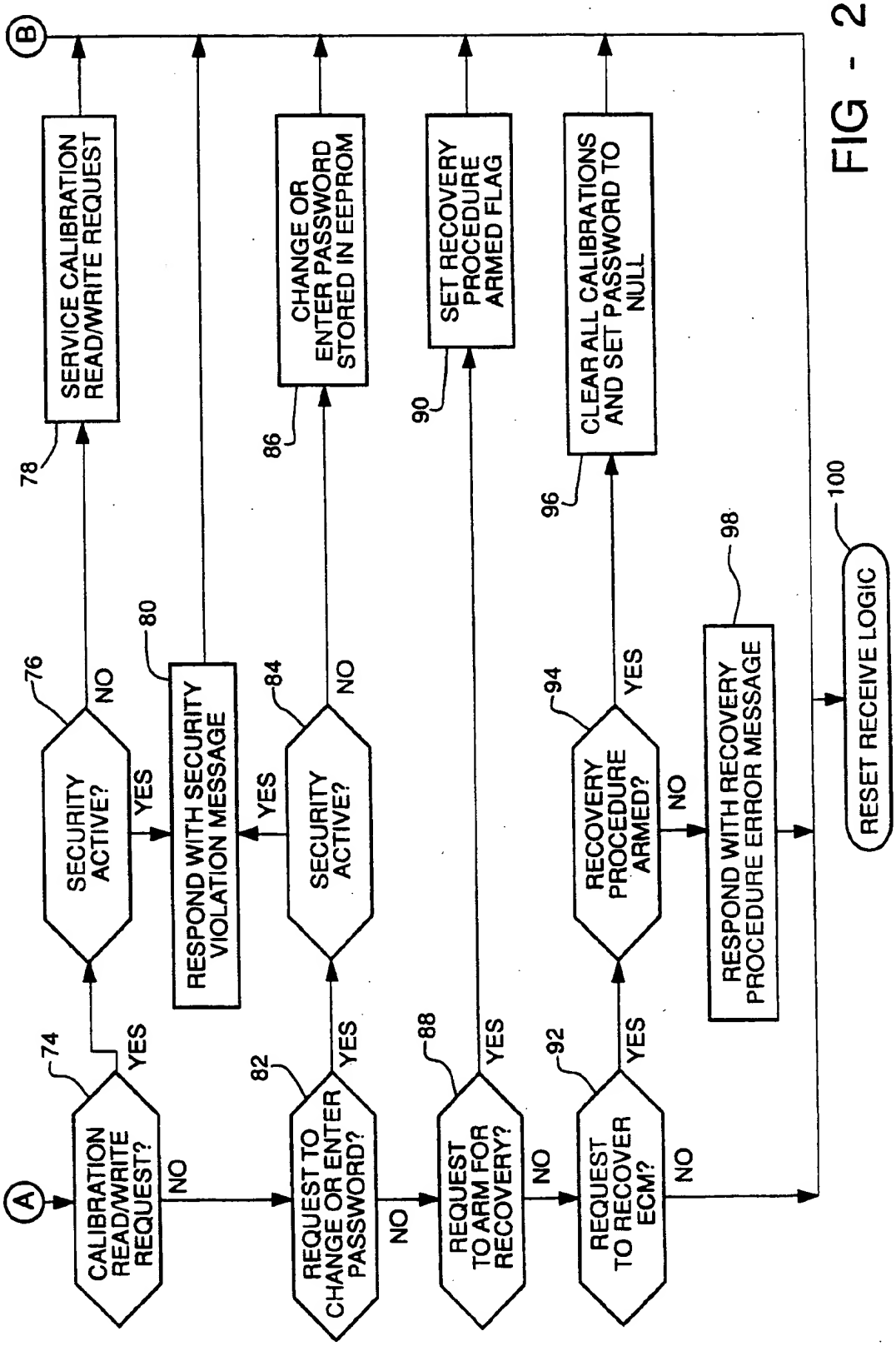

A
B
CALIBRATION READ/WRITE REQUEST?
74
YES
NO
SECURITY ACTIVE?
76
YES
NO
SERVICE CALIBRATION READ/WRITE REQUEST
78
RESPOND WITH SECURITY VIOLATION MESSAGE
80
REQUEST TO CHANGE OR ENTER PASSWORD?
82
YES
NO
SECURITY ACTIVE?
84
YES
NO
CHANGE OR ENTER PASSWORD STORED IN EEPROM
86
REQUEST TO ARM FOR RECOVERY?
88
YES
NO
SET RECOVERY PROCEDURE ARMED FLAG
90
REQUEST TO RECOVER ECM?
92
YES
NO
RECOVERY PROCEDURE ARMED?
94
YES
NO
CLEAR ALL CALIBRATIONS AND SET PASSWORD TO NULL
96
RESPOND WITH RECOVERY PROCEDURE ERROR MESSAGE
98
RESET RECEIVE LOGIC
100

FIG - 2B

SECURITY METHOD FOR PROTECTING ELECTRONICALLY STORED DATA

FIELD OF THE INVENTION

This invention relates to a method of operating an electronic control module and particularly to a method of securing protected data stored in such a module.

BACKGROUND OF THE INVENTION

Internal combustion engines used for automotive vehicles, marine applications and other uses are efficiently controlled by microprocessor based electronic control modules which receive information on operator commands, engine operation parameters and the like and issue control signals for spark timing and fuel control. Such controls are necessary to achieve fuel economy, optimum vehicle performance and compliance with emissions standards. The task of defining the optimum control parameters or calibrations requires the expenditure of much development time and expense for each engine type by the engine manufacturer, and results in an advantage over competitors lacking the optimized parameters.

The control modules are available to any engine manufacturer along with field service tools, calibrations installation tools and calibrations development tools. Heretofore each manufacturer could retrieve the valuable calibrations data from a module using standard tools and pirate the data for use in a similar competing engine. It is thus desirable for an engine manufacturer to safeguard the information installed in such a module. It is generally known to secure electronically stored data such as bank accounts by requiring passwords for access to the information. The same approach can be used for securing the calibration data. That is, a password must be entered in order to retrieve the sensitive data stored in the control module.

A disadvantage of password protection is that if the password becomes lost or forgotten, and it is necessary to access the memory to change the calibration, the module becomes useless. If the module hardware manufacturer becomes a repository of secret passwords, the chance of a security breach is increased, and it is a burden to both the repository and the user to retrieve the password. Moreover it is important for the sake of flexibility that the user create and install the password as required. Thus it is desirable to recover a unit having an unknown password without reducing the security.

Since modules for controlling various engine types have different embodied software, it is desirable to use different passwords for different types to increase security. The modules, however, are identical in physical appearance, and it sometimes occurs that the identity of a module is unknown and thus an authorized user has difficulty in determining which of several passwords to use. Thus it is desirable to establish a way for an authorized user to determine the correct password.

Because it is often necessary for service technicians to access information in the module to analyze engine operation, it is unrealistic to require a password for access to all data. Thus it is desirable to allow free access to that information which does not have to be protected while safeguarding other information.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to safeguard sensitive data in an electronic module by a password scheme while allowing the module to be recovered for use when the password is unknown. Another object is to assist in identifying the type of module to permit an authorized user to determine which password to use in accessing secure data. Still another object is to allow free access to non-sensitive data while safeguarding sensitive data.

A microprocessor-based engine control module suitable for use with a variety of engines contains software supplied by the module manufacturer for spark and fuel control. Specific control parameters or calibrations are developed for each type of engine by the engine manufacturer and loaded into a protected portion of non-volatile read/write memory such as an EEPROM. A password is also loaded into the same memory at a specific address. A part of the software is logic for password protection of the data in the protected portion of memory. Other data loaded into the module or generated by the module are not protected. At the time of manufacture the password address is loaded with a null password and the security is deactivated.

The module is coupled by a serial communication interface to calibrations development or calibrations installation tools which send digital messages to the module to load the calibrations and password, or to write other data to update the module, and read the calibrations, or other service tools to retrieve data useful in trouble shooting the module operation. The protection logic allows free access to unprotected data without the use of any password. Other communication of unprotected data is also allowed as between modules of two engines on the same watercraft or to an instrument panel.

Security is activated by writing a password to the EEPROM password address and removing the operating voltage from the module. This is accomplished by sending a code including the password via the serial communication interface. Once the password is installed and the security is activated, any request for protected data is refused. If the password is entered the security is deactivated for the remainder of the ignition cycle, and a subsequent request during the cycle for calibration data will then be honored, and if desired, the password itself may be changed.

If an incorrect password is entered, the logic will not deactivate the security and any subsequent password entry will not be entertained until the operating voltage is removed and then reapplied to the module. This feature makes it impossible to rapidly try a series of possible passwords to find the correct one. Instead, about two seconds is required to turn the voltage off and on and attempt another password. Because of this time delay feature it would take a single automated device up to 272 years to guess the password if an eight digit password is used. The module has no alternative paths to access the protected data so that there is no practical way for an unauthorized person to discover the data, even by disassembling the module.

Each type of engine control is preferably assigned a unique password and a corresponding identification number. Thus the manufacturer will list all the passwords in use and the corresponding identification numbers. The ID is entered into unprotected memory when the password is loaded. By requesting the ID from the module, the authorized user can determine the correct password from the password list.

If a password cannot be determined from the ID and the password list and is otherwise unknown, the module can be recovered for use by requesting recovery. This is done in two steps to prevent accidental recovery. The first step requires an arming request and the second step requires a recovery request. The protection logic responds to the two requests by

first erasing all the protected data and then deactivating security. Then the module may be reloaded with calibration data, a new password and corresponding ID.

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other advantages of the invention will become more apparent from the following description taken in conjunction with the accompanying drawings wherein like references refer to like parts and wherein:

FIG. **1** is a block diagram of an engine control module suitable for use with the method of the invention; and

FIGS. **2A** and **2B** comprise a flow chart representing the security program embedded in the module of FIG. **1** according to the invention.

DESCRIPTION OF THE INVENTION

Referring to FIG. **1**, an engine control module **10** is supplied with operating voltage through a vehicle ignition switch **11**. The module **10** is a single semiconductor chip which includes a microprocessor unit (MPU) **12**, an analog-to-digital converter (ADC) **14**, a read-only-memory (ROM) **16**, a random access memory (RAM) **18**, an electrically erasable programmable read-only-memory (EEPROM) **20**, an engine control unit (ECU) **22**, and a serial communication interface (SCI) **24**. The MPU **12** may be a microprocessor model MC-6100 manufactured by Motorola Semiconductor Products, Inc. Phoenix, Ariz. The MPU **12** communicates with the rest of the system by an 8 bit bi-directional data bus **25**.

The ROM **16** contains the program steps for operating the MPU **12**, and tables and non-sensitive constants used in determining engine fuel and ignition control parameters. The ROM **16** also contains security logic which is used to prevent unauthorized access to sensitive data stored in the EEPROM **20**. The EEPROM has an address for a password, and addresses for sensitive data, particularly calibration parameters, as well as addresses for non-sensitive data. The RAM **18** temporarily stores data which may be read from various locations determined in accord with the program stored in the ROM. Discrete inputs are directly coupled to the MPU and analog Inputs from operator controls or engine sensors are received at the ADC **14** and digitized for use by the MPU. The ECU **22** produces module outputs to the engine for fuel control and spark control.

The SCI **24** receives messages from and responds to external tools such as field service diagnostic tools **26**, calibration burn-in tools **28** and calibration development tools **30**. The diagnostic tools **26** are used by service technicians to trouble shoot engine operation, and request and receive non-sensitive information such as engine speed. The calibration development tools **30** are used by engineers in the development of calibration data which, when finalized, is saved in a file which is given to the engine manufacturer. The calibration burn-in tools **28** are used by the engine manufacturer to enter the file of calibration data into the EEPROM. The tools issue a number of digital codes which are interpreted by the module as requests or commands to read data from the module, write data into a memory location, or other instructions. The messages are routed to the security logic program which filters the messages, passing those dealing with non-sensitive data, and evaluating whether other messages should be honored. The SCI also handles communication with other on-board devices.

It is well known to provide an ECM with a bootstrap mode which is permitted by a special boot ROM area which manages the downloading of a program into RAM via the SCI, and access to calibration data can be obtained. The module may have architecture which provides the bootstrap capability, but it also has a security flag preventing such operation. The security flag is burned into the chip at the time of manufacture to positively prevent access to the calibration data. Further since the protection program and the EEPROM containing the protected data are on the same chip, it is not possible to physically remove the EEPROM from the protected environment to read the memory contents.

Upon manufacture of the module chip the password by default is set at FFFFFFFF. The program recognizes two password states, 00000000 and FFFFFFFF as null passwords and the security is deactivated for that condition. Thus the module is in condition to be loaded with secure data, a password and ID number by the engine manufacturer.

The security logic program is represented by the flow chart of FIGS. **2A** and **2B**. In the following description, numerals in angle brackets <nn> refer to functions in flow chart blocks bearing the corresponding reference number. When an incoming message is received from a tool or on-board device **<40>**, the logic examines the message code to determine the response to be sent via the SCI. If the request is for non-secure data from an on-board device such as an instrument panel or a twin engine **<42>** the module responds by providing non-secure on-board data **<44>**. If the message is a request for field service data **<46>**, then that data is supplied in response **<48>**. All other requests affect the security of the protected data. The program checks to determine whether a null password is present **<50>** and if so, the security is deactivated **<52>**. Next the program checks for a request for an ID number **<54>**. If ID is requested it is supplied to the requesting tool **<56>**.

When the message is an attempt to enter a password **<58>**, it is first determined if a previous attempt has been made during the same ignition cycle **<60>** and if so an error message to that effect is given in response **<62>**. If there was no previous attempt and the password is incorrect **<64>** a password incorrect message is given **<66>** and an "attempt" flag is set for subsequent use by block **60**. The flag will be cleared when the ignition cycle ends, i.e., when operating voltage is removed from the module. This feature imposes a time delay which defeats a brute force attempt to gain access by rapidly trying all possible passwords. If the password is correct **<64>** the security is deactivated **<70>** and a password correct message is returned **<72>**. The security will remain deactivated until the operating voltage is removed to allow free access to the protected data and the password.

If a calibration read or write message is received **<74>** and the security is not active **<76>** the request will be serviced **<78>**, and if security is active a security violation message will be sent **<80>**. Similarly, if a request to change the password or initially enter a password is received **<82>**, and security is not active **<84>** it will be honored **<86>** and if security is active the violation message will be sent **<80>**. Thus changes in protected data are easily made by one who has the correct password. When a password is entered the security is activated when the operating voltage is removed from the module.

If a password has been lost and it is necessary to change the calibration of the module, and even the report of the ID number does not yield a listed password, the unit is recovered by first requesting to arm for recovery. If such a request

is received <88> a recovery procedure "armed" flag is set <90>. Then if a request to recover message is received <92> and the "armed" flag has been set <94>, the calibrations are cleared from the EEPROM and the password is set to null <96>. If the "armed" flag is not set a recovery procedure error message is sent <98>. If the message does not fit any of the above categories or a response to a message has been made, the logic is reset to receive another message <100] at block 40.

It will thus be appreciated that the control module is furnished with a very high degree of security of data which must be protected while affording the user manufacturer ease of access to load or change the calibrations and even to reuse a module for which the password is unknown.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. In an electronic control module for an internal combustion engine comprising a microprocessor, a ROM containing a controller program and a data security program and a non-volatile memory having data protected addresses including a password address, means for reading and writing protected data to the data protected addresses, and communications means for receiving and sending messages; the method of maintaining secrecy of protected data comprising the steps of:

activating security by writing a password into the data protected password address;

when security is activated, rejecting messages requesting protected data;

deactivating security by supplying a correct password;

complying with requests for reading or writing protected data when security is deactivated;

entering a command for access to protected addresses irrespective of the state of security to recover said electronic control module without use of a password; and

responding to the command by erasing the protected data and then complying with the command, thereby ensuring the secrecy of the protected data.

2. The electronic control module of claim 1 wherein:

the step of entering a command for access comprises requesting release of security protection; and

the step of responding to the command comprises erasing the protected data and writing a null password.

3. The electronic control module of claim 1 wherein:

the step of entering a command for access comprises requesting release of security protection;

the step of responding to the command comprises erasing the protected data and writing a null password; and

reactivating security by writing new data and a new password to protected addresses.

4. The electronic control module of claim 1 wherein the step of entering a command for access comprises the steps of:

first, entering a request to arm for recovery; and

second, entering a request to recover the module.

5. The electronic control module of claim 1 wherein the steps of entering a command and responding to the command comprise:

entering a request to arm for recovery;

responding to the request by arming the module for recovery;

then entering a request to recover the module; and

responding to the command request to recover by erasing the protected data and deactivating security.

6. The electronic control module of claim 1 including the method of changing the password comprising the steps of:

requesting a change of password when security is deactivated;

supplying a new password; and

entering the new password at the password address.

7. The electronic control module of claim 1 wherein a control module has installed therein one of several sets of protected data wherein each set has a corresponding unique password and a corresponding unprotected identification number, wherein the method includes:

requesting the identification number of the installed set of protected data;

responding with the identification number corresponding to the set of protected data; and

entering the password corresponding to the identification number, whereby the security is deactivated.

8. The electronic control module of claim 1 further comprising the steps of:

energizing the module by supplying an operating voltage;

rejecting any attempt to enter an incorrect password; and

after rejecting such an attempt, rejecting any further password entry attempts while the operating voltage continues to be supplied.

9. The electronic control module of claim 8 wherein a correct password is accepted after a rejection by the steps of:

removing and then reapplying the operating voltage; and

then deactivating security by entering a correct password.

* * * * *